INTERNATIONAL ACADEMIC RESEARCH JOURNAL OF SOCIAL SCIENCE

IARJ INTERNATIONAL ACADEMIC RESEARCH JOURNAL

ISSN :2289-8441

International Academic Research Journal of Social Science

Journal homepage: www.iarjournal.com

Optimizing Cost-Effective Travel Modes between Two Destinations Using Dijkstra Approach

Wardah Mohd Nor[1], Noraini Noordin[2]

[1,2]Universiti Teknologi MARA Perlis, Arau 02600, Malaysia

Corresponding email: norainինoordin@perlis.uitm.edu.my

Article Information

Keywords
Shortest path problem, Dijkstra algorithm, network model, time-dependent SPP, cost-dependent SPP

Abstract

Inability of students to identify the shortest path and minimum cost path in a journey has attracted the researcher's attention to propose several travel alternatives modes to overcome the problem. The aim of this research was to find the minimum total travel cost and shortest total completion time for a journey between two destinations. The time-dependent shortest path problem (TDSPP) and cost-dependent shortest path problem (CDSPP) were drawn separately. In addition, Dijkstra algorithm applied in finding the shortest paths was calculated in C programming. Interestingly, both time-dependent and cost-dependent shortest paths did not map out similar paths.

INTRODUCTION

Mobility of society has been made easy by transportation through land, air and water. In addition, movement of goods and services provides the basis to run the economy of a country (Nagurney, 2007). For a student, transportation provides mobility to travel between two destinations. In this study, a network model was used to represent the journey between two destinations that involves multiple types of transportation.

This study aimed to identify minimum total travel cost and shortest total completion time for travel between these two destinations. Dijkstra has the ability to puzzle out any problem effectively as it is a simple, well-known algorithm for shortest path problem (SPP). Therefore, this study has chosen to use it in finding the shortest cost-efficient path between two destinations, in terms of time and money.

LITERATURE REVIEW

This problem was divided into two; time-dependent network model and cost-dependent network model.

Network Diagram

A network diagram describes the relationship between nodes and edges, direction of edges, as well as cost of nodes and edges. It is an important category of mathematical program that can be applied to numerous field such as communication networks, social networks and scientific collaboration networks (Lloyd and Valeika, 2012). It can also be modelled as networks optimization problems such as transportation problem, critical path, shortest path, minimum cost flow problems, and many more. In this study, a network model was used to find the shortest path from UiTM Perlis to Johor Bahru (JB) with the least time and fare.

The Shortest Path Problem (SPP)

Kamiński et al. (2011) have defined the SPP to be the problem of finding a path between two nodes in a graph that will minimize sum of the weights of its constituent edges. In particular, SPP was used in this research to find the best combination of transportation modes that would give shortest time and minimum cost between each pair of nodes. Delay time is represented by d_{ij} which denotes travel time needed from node *i* to node $j, (i, j) \in A$. The delay time associated is considered as time-dependent as it depends on when the travel started. In other word, the function $d_{ij}(t)$ returns time to travel from *i* to *j* when leaving *i* at time *t*. Therefore, the traveller will arrive at node *j* at time $t + d_{ij}(t)$ (Bérubé et al., 2006). In this research, the delay function is restricted to deterministic discrete time problems, where the time on each arc is known and finite with certainty.

Dijkstra Algorithm

Dijkstra algorithm is one of the classical algorithms in SPP, specialized in single-source shortest path problem in a directed graph with nonnegative weight. Dijkstra is a simpler and faster version of Ford's algorithm. However, it was assumed that the weight functions are known ahead of time, are monotonic and do not change (Abbasi and Ebrahimnejad, 2011). It maintains array of provisional distance, d for each node. The size of the search space is $O(n^2)$ and $n/2$ nodes on the average. The path can be stopped as soon as all target nodes are reached (Murota and Shioura, 2014). According to Jasika et al. (2012), two sets, D for all extended nodes and U for unintended nodes are fixed in this algorithm. Set D will increase as the minimal weight node from U is updated. This procedure will be repeated until all nodes are in D.

METHODOLOGY

This research has chosen to analyze the journey between UiTM Perlis and JB which included three (3) common types of transportation: bus, train, and airplanes. Transportation agencies involved were Transnasional, Sri Maju, City Express, AirAsia, Malaysia Airlines (MAS), Malindo Air, and Keretapi Tanah Melayu (KTM).

Current Alternatives for Travel Routes out of UiTM Perlis

Travel route from UiTM Perlis to JB was divided into three phases based on the following conditions:

- From UiTM Perlis to activity port A , Arau, Kangar, or Alor Setar (AS),
- From activity port A to activity port B, AS,
- From activity port B to activity port C, Kuala Lumpur (KL), and
- From activity port C to activity port D, JB.

The Assumptions

Dijkstra algorithm was applied according to several assumptions: i) fixed starting point, namely UiTM Perlis; ii) fixed end point, namely JB; iii) varying price of flight tickets; and iv) transportation out of UiTM Perlis to first activity port was by taxi only.

Network Model Construction

The first step in finding the shortest path is to determine activities that are related to the project (Marasovic and Marasovic (2006). List of activities arranged in Table 1 enabled the construction of the precedence table for this study. 245 travel routes with different modes were possible to be selected.

After specifying all the activities involved and decision on precedence activities has been made, the activity-on-arc technique was used to construct the models as it was easier and commonly used in commercial software. Two models were constructed here for the Time-dependent shortest path problem (TDSPP) and Cost-dependent shortest path problem (CDSPP). Fig. 1 displays the network for TDSPP. For CDSPP, the weights were given in terms of money.

TABLE I
LIST OF ACTIVITIES

No.	Activity	No.	Activity
1	From UiTM Perlis to Arau by taxi.	12	From AS to KL by Sri Maju
2	From UiTM Perlis to AS by taxi	13	From AS to KL by Transnasional
3	From UiTM Perlis to Kangar by taxi	14	From AS to KL by City Express
4	From Arau to AS by KTM	15	From KL to JB by AirAsia
5	From Kangar to AS by Sri Maju	16	From KL to JB by MAS
6	From Kangar to AS by Transnasional	17	From KL to JB by Malindo Air
7	From Kangar to AS by City Express	18	From KL to JB by KTM
8	From AS to KL by AirAsia	19	From KL to JB by Sri Maju
9	From AS to KL by MAS	20	From KL to JB by Transnasional
10	From AS to KL by Malindo Air	21	From KL to JB by City Express
11	From AS to KL by KTM		

After specifying all the activities involved and decision on precedence activities has been made, the activity-on-arc technique was used to construct the models as it was easier and commonly used in commercial software. Two models were constructed here for the Time-dependent shortest path problem (TDSPP) and Cost-dependent shortest path problem (CDSPP). Fig. 1 displays the network for TDSPP. For CDSPP, the weights were given in terms of money.

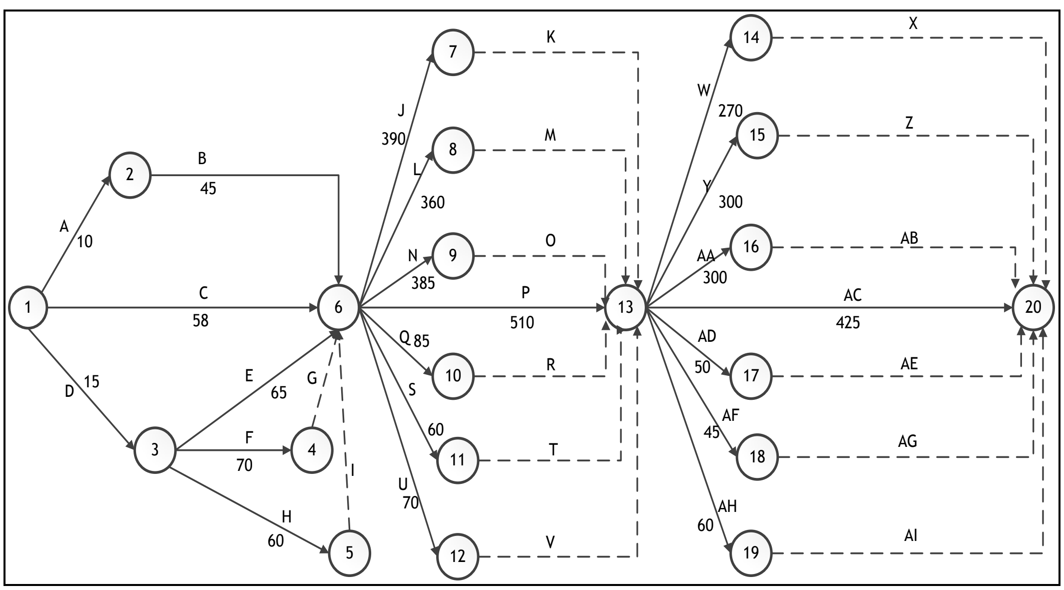

Fig. 1.
Time-dependent shortest path problem network diagram

In the final stage, time and cost for each activity were varied. From each possible route from UiTM Perlis to JB, the shortest path problem was calculated using Dijkstra algorithm in C programming.

Dijkstra Algorithm Implementation

Basic steps in Dijkstra algorithm used in this study followed Jasika et al. (2012).

1. Initialization. Set a source node, s and assign zero weighted value to it. Tag the source node as permanent $(0, p)$. Node is now the current node. For every other node, a distance value of ∞ is assigned and tagged as temporary (∞, t).
2. Update the distance value and current node designation. Let i be the index of the current node. Find another temporary tagged node that can be connected with i. Then update the distance values of (i, j) link as follows: $new\ d_j = \min(d_j, d_i + c_{ij})$ where c_{ij} is the cost of the link (i, j) as stated in network problem.
3. Determine a node j that has the smallest distance value d_j among all nodes that can be connected with i, such that, min $d_j = d*_j$.
4. Change the label of node $j*$ to permanent and designate this node as the current node.
5. Repeat from step 2, until the path reached node or all nodes have been permanently tagged, then the path has reached the end.

Objective Function

Let z be the total time for a travel. Eqn. 1 to 12 calculate the shortest time for each travel, as follows:

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}x_{ij} + \sum_{i=m+1}^{n}\sum_{j=m+1}^{n} d_{ij}x_{ij} + \sum_{i=m+n+1}^{p}\sum_{j=m+n+1}^{p} d_{ij}x_{ij} \tag{1}$$

where,

	p, m, n	=	total number of nodes at each stage respectively
	i	=	start point
	j	=	end point
	d_{ij}	=	time travel from city i to city j

subject to:

For nodes 1 to 6,
$$x_{ij} = \begin{cases} 1, & \text{if } x_{12}, x_{13}, \text{ or } x_{16} \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

For nodes 2 to 6,
$$x_{ij} = \begin{cases} 1, & \text{if } x_{26} \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

For nodes 3 to 6,
$$x_{ij} = \begin{cases} 1, & \text{if } x_{34}, x_{35}, \text{or } x_{36} \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

For nodes 6 to 13,
$$x_{ij} = \begin{cases} 1, & \text{if } x_{67}, x_{68}, x_{69}, x_{610}, x_{611}, x_{612}, \text{ or } x_{613} \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $x_{713}, x_{813}, x_{913}, x_{1013}, x_{1113}$, or x_{1213} are dummies.

For nodes 13 to 20,
$$x_{ij} = \begin{cases} 1, & \text{if } x_{1314}, x_{1315}, x_{1316}, x_{1317}, x_{1318}, x_{1319}, \text{or } x_{1320} \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where, x_{1420}, $x_{1520}, x_{1620}, x_{1720}, x_{1820}$, or x_{1920} are dummies

RESULTS AND DISCUSSIONS

Dijkstra Calculation by C programming

The time-dependent network and cost-dependent network model were run separately. Fig. 2 shows the input and result in C programming for the time-dependent network. Similar procedures were carried out for the cost dependent network.

As can be seen, the shortest path for TDSPP was was 0 1 5 10 12 17 19 which denoted the shortest path from UiTM Perlis to Arau by taxi, from Arau to AS by train, from AS to KL by Malaysia Airlines, and from KL to JB by

Malaysia Airlines. Total time displayed was 160 minutes or equal to 2 hours and 40 minutes. On the other hand, the shortest path for the CSDPP was 0 1 5 9 12 19 which was identified by the path from UiTM Perlis to Arau by taxi, from Arau to AS by taxi, from AS to KL by AirAsia, and from KL to JB by train. Total cost displayed was RM86.01

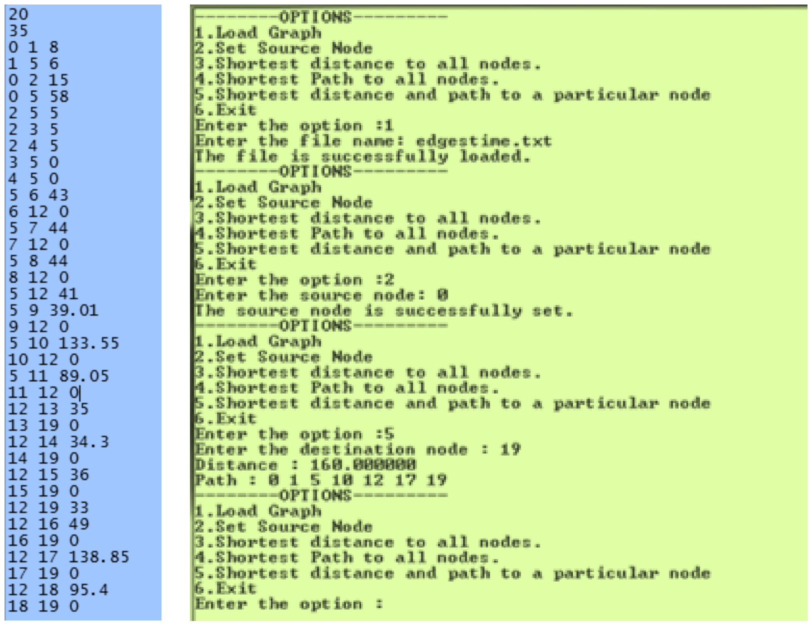

Fig. 2.
Input and result for time-dependent network

CONCLUSION

The SPPs in this study were modeled as TDSPP and CDSPP through a fixed sequence of nodes. Due to limited time for this study, both time-dependent shortest path and cost-dependent shortest path were treated separately. The cheapest path of RM86.01 was accomplished in a longer time. In contrast, the cost associated with TDSPP was RM286.40. Here, the travellers can choose either time-effective shortest path or cost-effective shortest path. With more modes of travel, the results would be more interesting. The focus of this study is applicable to many areas in the real world and not limited to transportation problems only. Further considerations may include converting to dynamic SPP by adding float time, or finding one shortest path for both time and cost.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknologi MARA Perlis for sponsoring this paper, and Transnasional, Sri Maju, City Express, AirAsia, Malaysia Airlines, Malindo Air, as well as KTM for allowing access to their data.

REFERENCES

Abbasi, S., & Ebrahimnejad, S. (2011). Finding the Shortest Path in Dynamic Network using Labeling Algorithm. *International Journal of Business and Social Science, 2*(20).

Bérubé, J.-F., Potvin, J.-Y., & Vaucher, J. (2006). Time-dependent shortest paths through a fixed sequence of nodes: application to a travel planning problem. *Computers & Operations Research, 33*(6), 1838-1856. doi: http://dx.doi.org/10.1016/j.cor.2004.11.021

Jasika, N., Alispahic, N., Elma, A., Ilvana, K., Elma, L., & Nosovic, N. (2012, 21-25 May 2012). *Dijkstra's shortest path algorithm serial and parallel execution performance analysis.* Paper presented at the MIPRO, 2012 Proceedings of the 35th International Convention.

Kamiński, M., Medvedev, P., & Milanič, M. (2011). Shortest paths between shortest paths. *Theoretical Computer Science, 412*(39), 5205-5210. doi: http://dx.doi.org/10.1016/j.tcs.2011.05.021

Lloyd, A. L., & Valeika, S. (2012). Network models in epidemiology: an overview *Complex Population Dynamics* (pp. 189-214): WORLD SCIENTIFIC.

Marasovic, J., & Marasovic, T. (2006, Sept. 29 2006-Oct. 1 2006). *CPM/PERT Project Planning Methods as E-Learning Optional Support.* Paper presented at the Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006. International Conference on.

Murota, K., & Shioura, A. (2014). Dijkstra's algorithm and L-concave function maximization. *Mathematical Programming*, *145*(1-2), 163-177. doi: http://dx.doi.org/10.1007/s10107-013-0643-2

Nagurney, A. (2007). Mathematical Models of Transportation and Networks.