Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 2, 2021, page 213-221

Maximum Reliable Path Based on Traveling Salesman Problem Using Floyd-Warshall Algorithm

[1]Nor Arina Yasisman and [2]Wan Rohaizad Wan Ibrahim

[1,2]Department of Mathematical Sciences
Faculty of Science, Universiti Teknologi Malaysia,
81310 Johor Bahru, Johor, Malaysia.

e-mail: [1]norarina@graduate.utm.my, [2]wrohaizad@utm.my

Abstract Traveling salesman problem is an algorithm to find the shortest path between set of points or places to be visited. This algorithm was proposed by two Mathematicians in 1800s which are W.R Hamilton and Thomas Kirkman. The method used to solve the maximum reliability by using the Floyd-Warshall algorithm. The formulation of this problem by comparing all possible paths and choose the smaller distance by comparing the original and new distance. The results will be simulated by computer programming using Microsoft Visual C++.

Keywords Traveling Salesman problem; Maximum reliable path; The Floyd-Warshall algorithm; Computer Programming.

1 Introduction

The background of the problem is when a salesman wants to travel through 15 cities in Johor to sell goods. He wants to travel through the cities by choosing the most reliable path. The problem will be solved by using the Floyd-Warshall algorithm and simulation of the result using Microsoft Visual C++.

The Floyd-Warshall algorithm is one of the methods that can be used in finding the vertices and edges with minimum weight either in a directed or undirected graph. Thus, the Floyd-Warshall algorithm will be improved to solve the most reliable path in the traveling salesman problem.

Based on relevant information obtained from each node and my modeling the calculation, the most reliable path can be find using The Floyd-Warshall algorithm. Factors such as distances, time consuming, traffic congestion or weather conditions can influence the reliability of the path chosen.

The objectives of this research are to study the Floyd-Warshall algorithm and making comparisons between this algorithm, to determine the maximum reliable path by using the Floyd-Warshall algorithm and simulate the result by computer programming using MS Visual C++.

2 Literature Review

This algorithm is an example of dynamic programming, splits the problem into smaller subproblems, then combines the answers. All the shortest distances between any two vertices can be calculated in V^3, where V is the number of vertices in a graph.

2.1 The Floyd-Warshall algorithm

The Floyd-Warshall algorithm is one of the examples of the shortest path algorithm which computes the shortest path between all pairs of nodes. It can find the shortest paths in a weighted graph with positive or negative edges of the weight. The time complexity for this algorithm is O (V^3) where V is the vertices. This algorithm has behavior with negative cycles. A negative cycle is a cycle with the negative sum of the value of the edges.

2.2 The Bellman-Ford algorithm

The Bellman-Ford Algorithm is one of the algorithms used to find the shortest path problem from a single source of vertex to another vertex. This algorithm can solve the graphs although some of the edges of the weights are negative values.

2.3 The Dijkstra's algorithm

This model computes the shortest-path problem for any weighted, directed non-negative weight graph. It can solve the graphs which consist of cycles; however, negative weights will lead to incorrect results. Dijkstra's algorithm can work properly, because all the edges are non-negative, and always chooses the vertices with the least short-path estimate. Dijkstra's algorithm requires a priority queue at each of a number of network nodes, in N iterations.

2.4 Comparison between The Floyd-Warshall, The Bellman-Ford and The Dijkstra's algorithm.

Table 1: The comparison between three methods that can be used to solve shortest path problems.

The Floyd-Warshall algorithm	**The Bellman-Ford Algorithm**	**The Dijkstra's Algorithm**
Computes shortest distances between every pair of the vertices.	Computes the shortest path only from a single source.	Computes the shortest path only from a single source.
Works for the shortest path with negative edges.	Works for the shortest path with negative edges.	Aims for the graphs with non-negative weight of nodes.
The space complexity is $O(V^2)$	The space complexity is $O(V)$ which is linear	The space complexity is $O(V)$ which is linear

3 Methodology

This chapter will cover the methodology of the Floyd-Warshall algorithm, pseudocode, flowchart that is used to achieve the objectives of the research. The overview of the case study will be discussed in this chapter.

3.2 Modelling formulation

The shortest path between all nodes can be calculated using the Floyd-Warshall algorithm using the basic idea which has three steps:

Step 1: Find two vertices from the nodes and put each vertex in the nodes into these two points.

Step 2: Compare the new distance with the original distances and take the smaller distance as the new shortest distance.

Step 3: Construct the n matrix $k(1), k(2), \dots, k(n)$ sequentially by looping iteration. Each element in the last matrix $k(n)$ represents the shortest distance between the two points.

Step 4: Calculate the minimum distance of one point to other points by getting the sums of these elements in each line. One or more best locations can be found by comparing these summations.

3.3 Pseudocode

The pseudocode of this algorithm used two equations which are for the positions, *P* and the dimensions, $\widehat{D}_1$. Then, the value of $P_n(O_iO_j)$ will be initialized. The next step is, the value of $P_{1(O}iOj)$ will be calculated using the equation (1). The value of $\widehat{D}_1$ will be calculated using the second equation, (2). This algorithm used the loop for to repeat the process. The value of *n* will depend on the number of cities which consists of 15 cities in Johor. The whole process will repeatedly calculate the value and will stop when the value of *i* and *j* will be equal to the value of *n*.

$$P_{(O_iO_j)} = f\ (\ P_{1(O_iO_j)}, P_{2(O_iO_j)}, \dots, P_{n(O_iO_j)}) = P_{1(O_iO_j)}\ P_{2(O_iO_j)}\ P_{3(O_iO_j)} \dots P_{n(O_iO_j)} \qquad (1)$$

$$\widehat{D}_1 = \lceil P_{11}\ P_{12}\ P_{13}\ P_{21}\ P_{22}\ P_{23}\ P_{31}\ P_{32}\ P_{33}\ \rceil \qquad (2)$$

Initialize $P_{1(O_iO_j)}$, $P_{2(O_iO_j)}$, $P_{3(O_iO_j)}$, … , $P_{n(O_iO_j)}$
while (the task still in progress)

Calculate $P_{1(O_iO_j)}$ by equation (1)

Calculate $\widehat{D}_1$ with equation (2)
for (k=1; k<=n; k++)
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
if (the path with max reliability through node O_m)

then $O_{i'j'm} = O_{i,m,m-1} + O_{m,j,m-1}$

end if

if (the update time is up)

update $P_{1(O_iO_j)}$, $P_{2(O_iO_j)}$, $P_{3(O_iO_j)}$, … , $P_{n(O_iO_j)}$

3.4 Flowchart

This flowchart shows the process of the simulation for the traveling salesman problem using Microsoft Visual C++.

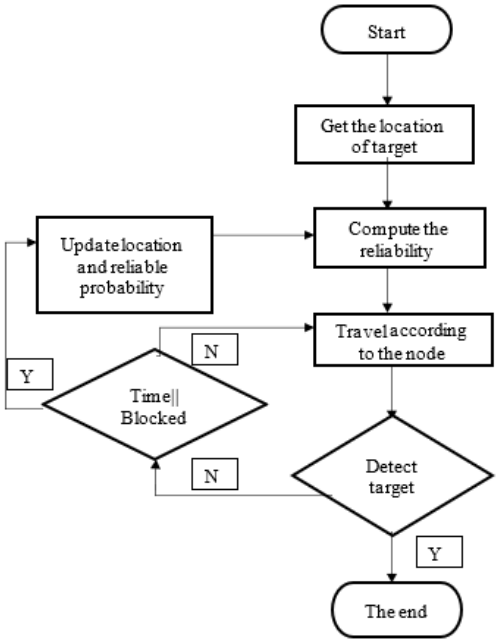

Figure 2: Procedure figure of the most reliable path planning

3.5 Data

A salesman is expected to make a round trip to all of the other cities in Johor to sell goods before returning home. The chosen route must have the maximum reliability. The data set is made up of 15 cities in Johor.

Let C_1= Bandar Penawar, C_2= Mersing, C_3= Endau,
C_4= Skudai, C_5= Kulai, C_6= Kluang, C_7= Simpang Renggam,
C_8= Ayer Hitam, C_9= Yong Peng, C_{10}= Batu Pahat, C_{11}= Segamat,
C_{12}= Muar, C_{13}= Gemas, C_{14}= Johor Bahru, C_{15}= Tangkak.

Figure 3.5.1 depicts a map of the 15 cities in Johor. To sell goods, the salesman will travel from one location to another. The chosen route will be the most dependable. Each city's destination will be represented by xy -coordinates. The Cartesian coordinates will be converted into Windows coordinates for simulation using Microsoft Visual C++.

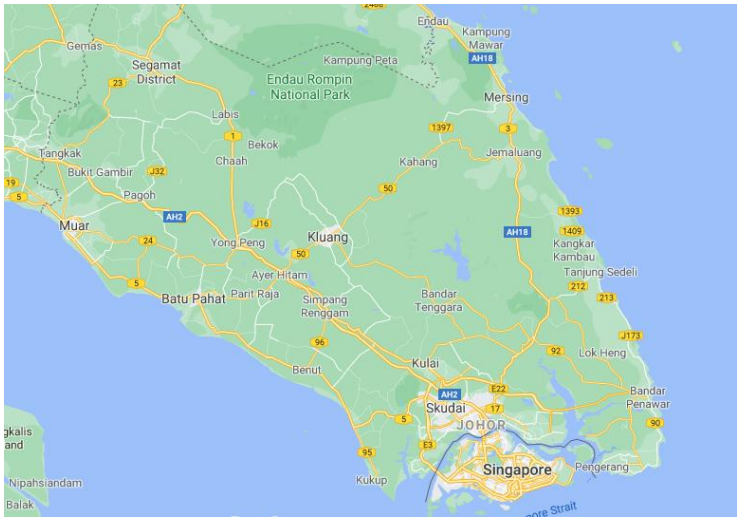

Figure 3.5.1 Map of the 15 cities in Johor.

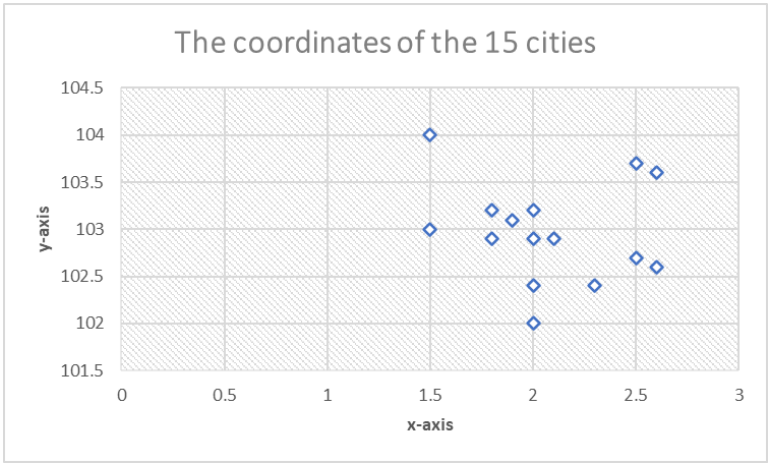

Figure 3.5.2 The cartesian coordinates of the 15 cities are plotted.

Table 2: The xy-coordinates of the 15 cities

x-coordinate	**y-coordinate**
1.5	104.0
2.5	103.7
2.6	103.6
1.5	103
2.1	102.9
2.0	103.2
1.8	103.2
1.9	103.1
2.0	102.9
1.8	102.9
2.5	102.7
2.0	102.4
2.6	102.6
2.0	102
2.3	102.4

3.6 Conversion of Cartesian coordinates to Windows coordinates

Table 3.6 The pixel coordinates of the 15 cities in Johor.

City	x-coordinate	y-coordinate
Bandar Penawar	749.8	328.7
Mersing	528.5	190
Endau	52.9	199.8
Skudai	520.9	543.2
Kulai	199.5	192.6
Klung	413.9	321.6
Simpang Renggam	85.3	148.1
Ayer Hitam	199.4	321.5
Yong Peng	306.9	165.4
Batu Pahat	192.4	543.3
Segamat	635.5	543.4
Muar	45.9	321.4
Gemas	749.9	250
Johor Bahru	84.9	436.1
Tangkak	521.1	240

4 Result

The expected outcomes of the maximum reliable path for the traveling salesman problem using the Floyd-Warshall algorithm will be discussed in this chapter. The factors that will be used to determine the most reliable path are distances travelled, time consumed or impact of the weather to travel from one place to another.

The result obtained will be simulated using MS Visual C++. The reliability of each city will be set randomly using programming. The figure below shown the output of the simulation that the most reliable path that will be used for the salesman to travel from one city to another city.

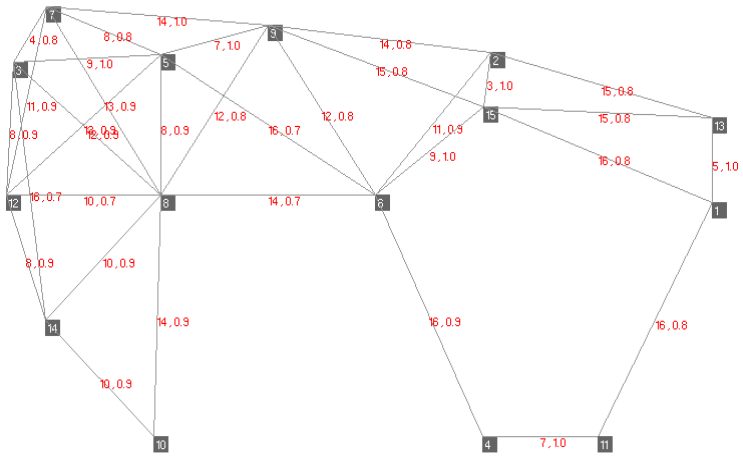

Figure 4.1 The output of the model

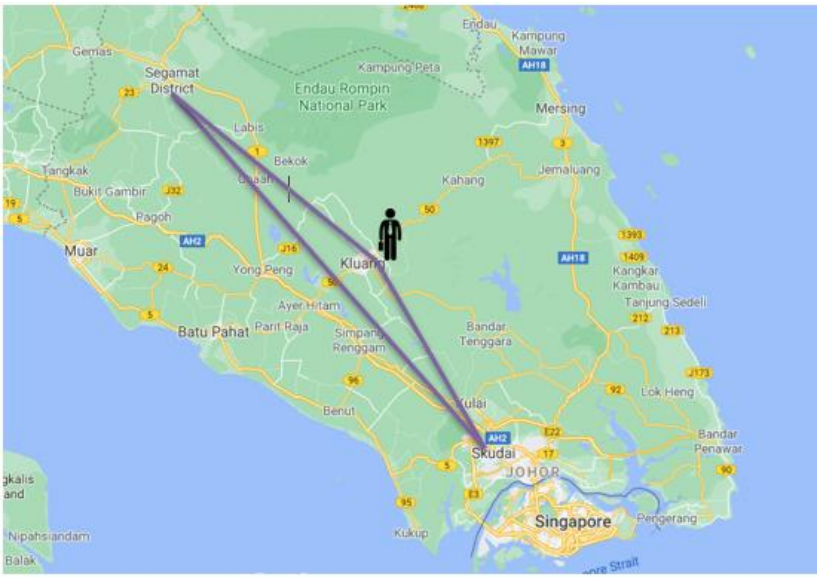

Figure 4.2 The reliability of the chosen path from Kluang to Segamat in real mapping

5 Conclusion

Based on the modelling of the most reliable path using the Floyd-Warshall algorithm, the travelling salesman problem can be solved. The simulation using MS Visual C++ helps in determining the most reliable path if the two cities are chosen. The implication from this study, the simulation of the result will help to understand the concept of sequence planning using the chosen algorithm.

The advantage of this study is to understand the concept and how the path planning can be important factors for the salesman to choose based on the reliability of the path. The limitation of this study is the path planning only based on the reliability, but other factors such as distance to travel and traveling cost should be included.

References

[1] Zhang B, Liu Y, Lu Q, Wang J. (2016). *A path planning strategy for searching the most reliable path in uncertain environments*. International Journal of Advanced Robotic Systems. doi:10.1177/172988

[2] Wang, Xiao Zhu. (2018). *The comparison of Three Algorithms in Shortest Path Issue*. International Conference on Advanced Algorithms and Control Engineering. doi:10.1088/1742-6596/1087/2/022011

[3] Hougardy, Stefan. (2010). *The Floyd-Warshall algorithm on graphs with negative cycle*s. Inf. Process. Lett.. 110. 279-281. 10.1016/j.ipl.2010.02.001.

[4] Vattai. (2016). *Floyd-Warshall in Scheduling Open Networks*. Procedia Engineering. 164(106-114). doi:10.1016/j.proeng.2016.11.598

[5] Mohammed. (2019). *Floyd Warshall Algorithm| DP-16.* GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/

[6] Sridharan T. (n.d). *Floyd Warshall Algorithm*. Medium.com. Retrieved from https://medium.com/dev-genius/floyd-warshall-algorithmf004a01ae40e

[7] Joshi V. (2019). *Comparison of Dijkstra's and Floyd-Warshall algorithms.* GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/comparison-dijkstras-floyd-warshallalgorithms/