Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 8, 2022, page 178-186

Applying Ideal Route Optimization for Traveling in Terengganu and Kelantan

Nurul Ika Mirlia Zarimi, Wan Rohaizad Wan Ibrahim
Department of Mathematics, Faculty of Science
Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

Abstract
Nowadays, tourist is interested in visiting attractive places that have their own uniqueness either for their job purpose or to spend their free time. However, the tourist still facing the difficulties in determining the optimal route to travel independently. These problems are referred as Travelling Salesman Problem (TSP). TSP is a problem where a person wants to travel between cities, but it should be started at the origin and returned back to the origin with the minimum distance. The main objective of this research is to determine the best route for traveller use to minimize the distance. Therefore, a solution is provided to solve TSP problem by using Tabu Search (TS) method because this method can produce optimum tourist route. Tabu Search enable traveller to find the optimal solution by visiting exactly once for node. The result shows that the optimal route is $Depot - 18 - 21 - 12 - 11 - 8 - 13 - 24 - 14 - 15 - 6 - 7 - 9 - 5 - 4 - 3 - 2 - 23 - 1 - 10 - 16 - 17 - 20 - 22 - 19 - Depot$ with the total distance $227.758km$. Microsoft Visual is used to assist in accomplishing the calculation involved.
Keywords: Travelling Salesman Problem (TSP); Tabu Search (TS).

1. Introduction

Operations research (OR) is a subfield of mathematical sciences that solves complex decision-making problems by applying methods from other mathematical sciences such as modelling, statistics and optimization. OR is frequently essential to determine the extreme value of some real-world objective: the maximum (of profit, performance, or yield) or minimum (of loss, risk or cost).

Travelling is a fantastic way to learn new things in life. Some people travel to learn more, while others travel to escape their daily lives. Whatever the reason, travelling allows us to explore the world beyond our imagination and indulge in a variety of activities. Travelling has become more convenient as technology and transportation have advanced. The goal of travelling is to maximize the experiences at the travel destination rather than spending time googling places to visit or deciding what to do.

Travelling Salesman Problem (TSP) requires finding a tour in the graph that visits each node exactly once so that the sum of the cost of the tour’s edges or arcs is as minimal as possible. TSPs are used to represent wide range of managerial issues, particularly in logistics and distribution. There are 2 types of TSP: symmetric TSP and asymmetric TSP. Symmetric TSP indicates that the distance between two cities is the same in both directions. Path between the asymmetric TSP may or may not exist in either direction or the distances may be different. Asymmetric TSP forms an undirected graph, whereas symmetric TSP forms a directed graph.

In this study, TSP problem will be solve using a Tabu Search. Tabu Search (TS) is a meta-heuristic search algorithm that uses the concept of short-term memory to avoid getting stuck in a local minimum. It has been used in variety of applications, one of which is the TSP. TS is a method for solving combinatorial optimization problems in order to find optimal solutions. The places chosen are Terengganu and Kelantan where the tourists will visit each of the cities in Terengganu and Kelantan.

The objectives of the research are: (1) to determine the best route for traveller use in order to minimize the distance and cost and (2) to apply TS model in selecting the route between fixed origin and destination.

2. Literature Review

2.1. Travelling Salesman Problem

Travelling Salesman Problem (TSP) is an algorithmic problem where it is an extension of Hamiltonian path problem which is to find the shortest route between a set of points and location that must be visited at least once. The salesman's goal is to minimize the travel cost, time and distance travelled (Wikipedia Contributor, 2020). A person will travel between cities where they must start from the city of origin and return back to the city of origin (Uwaisy et al., 2019).

There are several applications by the TSP which are logistics, manufacture of microchips, genetics, telecommunication, neuroscience, and others. TSP can be represented as undirected weighted graph. By Figure 1, the graph's vertices can be represented as cities, graph's edge represent as paths between one city to another city and the edge's weight represent the path's distance.

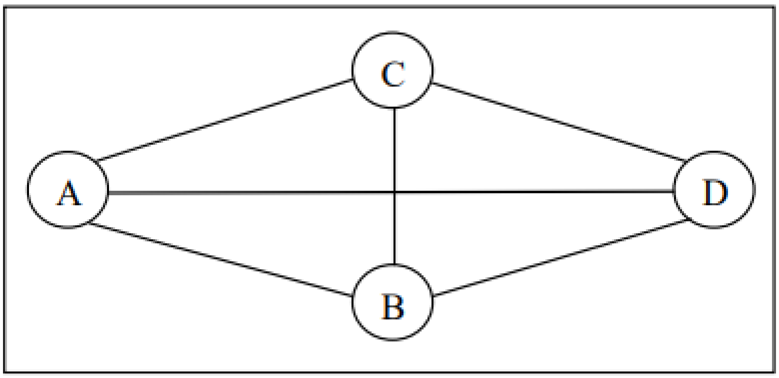


Figure 1 Undirected Weighted Graph

The number of feasible solutions is calculated by using the formula, $\frac{(n-1)!}{2}$ where after the starting point, there will be $(n-1)$ possible ways to visit the cities, $(n-1)$ for the next and so on. The denominator 2 indicates that there will not consider the routes to be counted twice (Brucato, 2013).

2.2. Heuristic Method

Heuristic methods are a strategy for locating a solution to a problem. It shorten the time it takes to find a good solution (Mulder, 2021). But according to Ohri (2021), heuristic approaches are designed to be adaptive and are used to make sensible decisions, especially when trying to find an ideal arrangement that is either impractical or impossible to achieve, or when dealing with complex data. Heuristic are mental shortcuts that can help the cognitive load to draw a conclusion fast.
According to Ohri (2021), there are 5 steps to solve heuristic method:

1) Determine the nature of a problem.
2) Create a hypothesis.
3) Data collection.
4) Analyse the data.
5) Make a conclusion.

Heuristic method also has their own advantages and disadvantages which are the advantages are it can provide designers with quick and moderately reasonable feedback and also can get feedback early in the design process while for disadvantages are successful application of the heuristic necessitates prior experience and knowledge and also qualified convenience specialists are hard to come by and can be very expensive (Ohri, 2021).

2.3. Tabu Search

Tabu Search (TS) is a metaheuristic for optimising model parameters that is widely utilised (Liang, 2020). TS is frequntly thought of as a way to include memory structures into local search algorithms. Because local search has many limitations, TS was created to address many of them. Its aim to solve general optimization problems by reaching a solution that is close to a global optimum.

TS has been use in variety of fields such as resource planning, telecommunications, financial analysis, scheduIng, logistics and others (wikipedia contributor, 2021). Also another example of TS can be solve are TSP, minimum spanning tree (MST) and vehicle routing problem (VRP) (liang, 2020). The advantages of TS are by selecting non-improving solutions, it is possible to avoid local optimums, Tabu List can be used to avoid reverting to prior solutions and cycles and also it can be used to solve both discrete and continuos problems.

The question posed by this problem is simple where there are given a list of cities, find the shortest route to visits every city. The serahc is begin with a preliminary solution, which can be created randomor using nearest neighbour technique. To generate new solutions, the order in which two cities are visited in a prospective solution is swapped. Total distance travelled between all the citis is used to compare how optimal one solution is to another. A solution is added to Tabu List if it is accepted into the solution neighberhood to avoid cycles that is constantly accessing the same set of solutions and to avoid becoming stuck in local optima.

TS has beend aapplied in many real-life applications such as Approximation of Shortest distance of a TSP (Rashid Khan& Asadujjaman, 2016), Application of Asymmetric TSP (Basu et al., 2017) and others. The successful implementation of TS has influenced to be apply in detemining the shortest route to travel.

2.4 Summary

In this section, the history of TSP, VRP , Heuristic Method, Tabu Search and Simulated Annealing have been discussed. From the previous study, TS have been used a lot in solving different types of TSP problems. The reason TS have been selecting to solve these problems is due to the success of implementation by the previous research and there is no study in determining the optimal route for travelling. This research applies TS methods to find the best route to travelling which can be assigned as an application of TSP. In the next chapter, it will present the further discussion of the TS method and algorithm.

3. Methodology

3.1. Tabu Search Elements

There are five steps to setup the algorithm for TS:

1. Inital Solution

   Any solution that meets the requirements for an acceptable solution is qualifies. In generation it, the shortest distance of the edges will be selcted (Rashid Khan & Asadujjaman, 2016).
2. Neighbourhood Structure

   Neighbourhood structure are functions that can be used to find other solutions by swapping two nodes in a solution. It must eliminate unnecessary and infeasible moves (Zhong, Wu, Li, & Ning, 2008).
3. Tabu List

   Tabu list are conditions that prevent the use of previously discovered solutions.
4. Aspiration Criteria

   Aspiration criteria aims to invalidate tabu status of neighbour if the solution leads to a better one. TS is too powerful, hence it may skip a good solution. Thus, aspiration criteria helps in canceling the tabu status of solution if it is deemed to provide a bether that current solution. (Rashid Khan & Asadujjaman, 2016).
5. Termination Criteria

   Ts algorithm is terminated based on the number of iterations and consecutive iterations specified by the user (Glover, 1990).

3.2. Travelling Salesman Problem Mathematical Formulation

TSP is the minimum Hamiltonian Cycle problem in a complete graph. Starting from a city n, travelling saleman need to visit each of the other city axactly once and return back to the origin city, covering minimum distance.
Euclidean distance from city i to city j is given by

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Where
n is the set of vertices(cities)
d_{ij} is the distance from city i to city j

The symmetric TSP model is given by

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \qquad (1)$$

Subject to:

$$\sum_{i=1}^{n} x_{ij} = 1, for\ j = 1,2, \dots, n \qquad (2)$$

$$\sum_{j=1}^{n} x_{ij} = 1, for\ i = 1,2, \dots, n \qquad (3)$$

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases} \qquad (4)$$

$$x_{ij} = 0, for\ i = j \qquad (5)$$

Where (1) is the objective function that minimize the distance travelled, (2) and (3) to make sure each city is visited once and visit all the cities, (4) give a value one if traveller travels from city i to city j and zero otherwise.

4. Results and discussion

4.1. Implementation using Microsoft Visual C++ 2019

A simple graphical user interface was implemented in this study to present the various cities, travel routes and computational findings for the travelling since it involves thousands of iterations, and it will take time to calculate manually. Figure 2 shows the graphical user interface of the program. The large field mark with the green line will display all the cities when the “read data” button is pushed. The display cities will be connected after the “set initial solution” button is triggered. The result will be display in the small field mark with the blue line. Each of the buttons has different functions and is simplified as follows:
Clear – to reset the program, clears the screen and all the variables used before.
Read Data – to generate data from imput.txt file into the coding.
Set Initial Solution – to generate an initial solution for the problem.
Compute -T.S. – to start the calculation of tabu search solutions for the problem.
Random – to generate random number of cities that can be specified the number by the user in the above “Random” button.

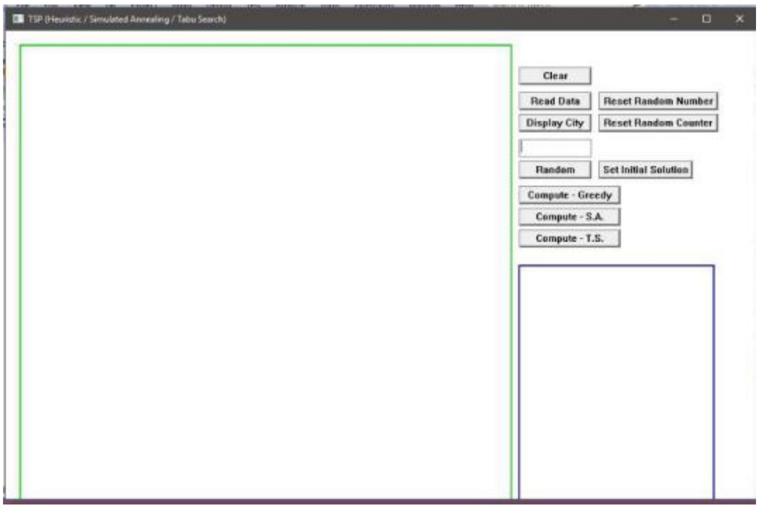


Figure ***Error! No text of specified style in document.****2*　　　*Graphical User Interface*

The simulated data are presented in Table 1 represents the location of the attraction's places in Terengganu and Kelantan. The first coordinates are depot and the following after the first coordinates will be marked as city $1,2,3, \ldots, n$. Once the "Read Data" button being pushed, all coordinates will be display in the output window. After that, hit the "Set Initial Solution" button to utilize the distance. This process is continuing until no further location is calculated. The results of the distance that have been computed are represented in Table 2.

In this TSP application, the goal is to travel the least distance via a specified number of cities, with the requirement that each city be visited once, starting from a depot and returning to the starting point. The study is conducted using the tabu search method, with Marang serving as both the starting and finishing points of the routes. The dataset includes all of the tourist attractions. The obtained solutions show the route taken.

Table **Error! Use the Home tab to apply 0 to the text that you want to appear here.**1　Simulated data to represent location of attraction places.

Cities	***x***	***y***
Depot	59	37
1. Pantai Bukit Keluang	26	30
2. Siti Khadijah Market	7	25
3. Wakaf Che Yeh	8	24
4. Inspirasi Kampung Aril	13	25
5. Jeram Mengaji	18	20
6. La Hot Spring	34	22
7. Kompleks Gua Ikan	28	1
8. Air Terjun Chemerong	73	18
9. Guillemard Bridge	17	14

10. Setiu Wetland	33	32
11. Kemaman Zoo & Recreational Park	94	23
12. ATV Tasik Puteri	75	27
13. Air Terjun Sekayu	62	24
14. Kenyir Lake	54	22
15. Gunung Sarut	45	24
16. Terrapuri Heritage Village	35	33
17. Surau Langgar Rindu	44	37
18. Pinehill Garden	62	36
19. Masjid Terapung Kuala Ibai	56	38
20. Terengganu State Museum	53	36
21. Jambu Bongkok Beach	72	36
22. Pasar Payang	53	38
23. Pantai Nami Bachok	8	30
24. Memorial Stone Inscription	60	28
25. Mini Zoo Kampung Tok Dir	57	35

25 cities will be generated at the beginning of the programme and stored in an array. This information is kept in a text file. The depot is indicated by the first set of coordinates, while the cities $1, 2, 3, \ldots, n$ are indicated by the points after that. The programme will read all the coordinates and display them in the output window when the "Read Data" button is clicked.

By choosing the closest outlet from the depot, the Tabu Search approach utilised in this study generates a preliminary solution. Without returning to the previously visited node, the traveller moves from the first node to the closest city using the same search approach. The traveller must return to the beginning place after arriving at the last city.

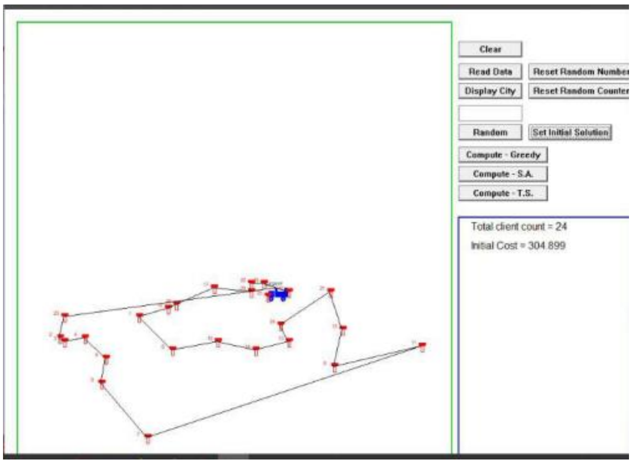


Figure 3 Initial Solution of Tabu Search

Figure 3 represent the initial solution generated by the program that will be used for Tabu Search. Total distance is km and the squnce of the initial solution is given by $Depot— 18 - 19 - 22 - 20 - 17 - 16 - 10 - 1 - 6 - 15 - 14 - 13 - 24 - 21 - 12 - 8 - 11 - 7 - 9 - 5 - 4 - 3 - 2 - 23 - Depot$.
This solution will be printed into the output file named “initialout.txt”

Table 2 Distance between city i to city j

Cities	Distance																									
	Depot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Depot	0	113	175	187	160	166	151	242	106	166	89	132	76	47	62	75	75	47	9.2	10	22	37	17	167	35	13
1	113	0	64	60	47	53	50	155	201	65	25	237	171	132	114	79	48	66	125	106	99	152	99	56	116	108
2	175	64	0	7.7	22	41	97	151	260	58	87	296	230	191	173	138	112	131	201	170	159	228	165	16	175	172
3	187	60	7.7	0	21	37	93	146	256	54	93	292	226	186	168	134	108	126	197	166	155	224	160	20	170	168
4	160	47	22	21	0	23	77	138	240	46	69	276	210	170	153	118	91	110	168	150	141	196	142	24	154	152
5	166	53	41	37	23	0	70	123	233	31	67	269	203	164	146	111	85	104	174	143	132	201	138	45	147	145
6	151	50	97	93	77	70	0	173	219	83	61	255	189	149	132	97	71	90	160	129	118	187	124	95	133	131
7	242	155	151	146	138	123	173	0	302	107	171	338	272	230	213	211	188	206	249	239	232	272	237	161	214	232
8	106	201	260	256	240	233	219	302	0	246	182	78	37	104	115	155	165	137	113	111	121	88	126	255	91	104
9	166	65	58	54	46	31	83	107	246	0	81	282	216	176	158	124	98	116	187	156	145	214	150	68	160	170
10	89	25	87	93	69	67	61	171	182	81	0	218	152	112	94	60	24	42	101	82	75	128	75	79	96	87
11	132	237	296	292	276	269	255	338	78	282	218	0	65	141	152	192	202	174	129	148	158	98	163	292	128	141
12	76	171	230	226	210	203	189	272	37	216	152	65	0	74	85	125	136	108	84	81	91	58	96	225	61	74
13	47	132	191	186	170	164	149	230	104	176	112	141	74	0	43	71	112	84	55	52	68	74	73	185	19	45
14	62	114	173	168	153	146	132	213	115	158	94	152	85	43	0	53	90	83	66	59	66	85	71	167	27	52
15	75	79	138	134	118	111	97	211	155	124	60	192	125	71	53	0	55	48	96	65	54	117	59	132	55	73
16	75	48	112	108	91	85	71	188	165	98	24	202	136	112	90	55	0	28	84	66	61	114	60	102	92	74
17	47	66	131	126	110	104	90	206	137	116	42	174	108	84	83	48	28	0	58	40	33	86	36	120	71	44
18	9.2	125	201	197	168	174	160	249	113	187	101	129	84	55	66	96	84	58	0	19	31	32	26	196	42	22
19	10	106	170	166	150	143	129	239	111	156	82	148	81	52	59	65	66	40	19	0	12	46	7.8	159	39	9.3
20	22	99	159	155	141	132	118	232	121	145	75	158	91	68	66	54	61	33	31	12	0	57	5.7	154	54	14
21	37	152	228	224	196	201	187	272	88	214	128	98	58	74	85	117	114	86	32	46	57	0	55	224	62	50
22	17	99	165	160	142	138	124	237	126	150	75	163	96	73	71	59	60	36	26	7.8	5.7	55	0	155	44	14
23	167	56	16	20	24	45	95	161	255	68	79	292	225	185	167	132	102	120	196	159	154	224	155	0	169	179
24	35	116	175	170	154	147	133	214	91	160	96	128	61	19	27	55	92	71	42	39	54	62	44	169	0	32
25	13	108	172	168	152	145	131	232	104	170	87	141	74	45	52	73	74	44	22	9.3	14	50	14	179	32	0

The output files for the TS method will be displayed in a format where the first text file column will show the iteration, the second column will show the most widely approved solutions, and the third column will show the best optimal solution. Following their presentation, the final iteration will show the routes' optimised order underneath. The routes shown are recommendations for the traveller to follow to reduce the distance.

In Figure 4, the first column represents number of iterations, while for the second column represent generated current solution and the third column represent optimal solution.

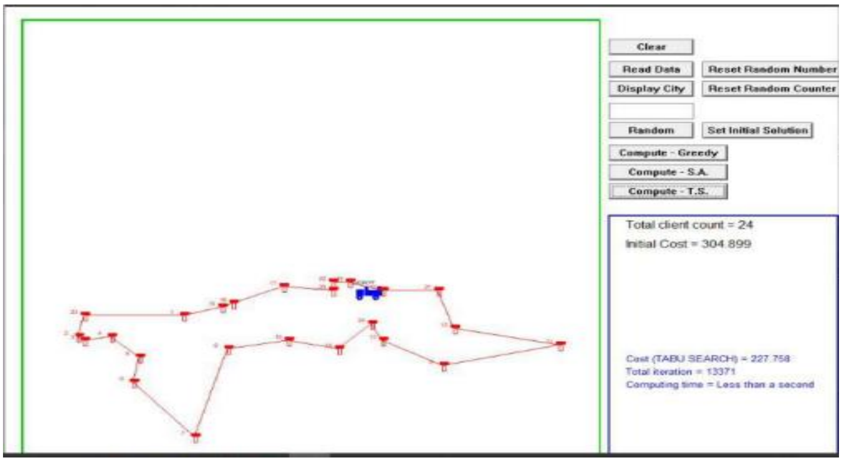
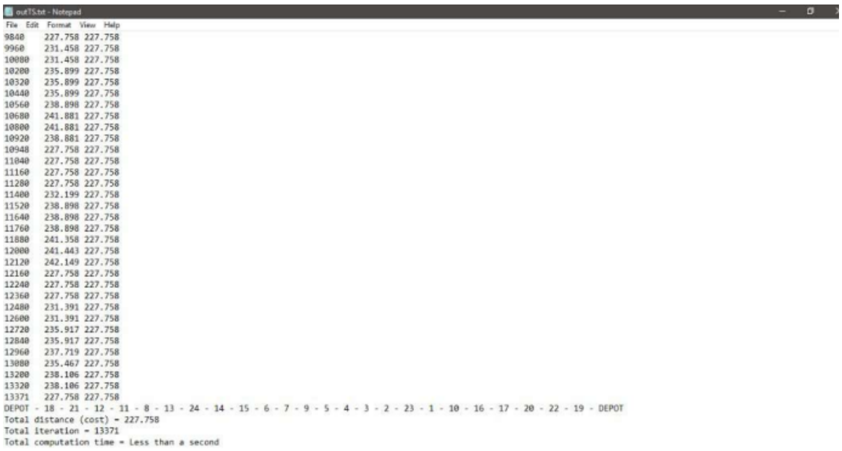


Figure 4 Output for Tabu Search



```
outTS.txt - Notepad
File Edit Format View Help
9840    227.758 227.758
9960    231.458 227.758
10080   231.458 227.758
10200   235.899 227.758
10320   235.899 227.758
10440   235.899 227.758
10560   238.898 227.758
10680   241.881 227.758
10800   241.881 227.758
10920   238.881 227.758
10948   227.758 227.758
11040   227.758 227.758
11160   227.758 227.758
11280   227.758 227.758
11400   232.199 227.758
11520   238.898 227.758
11640   238.898 227.758
11760   238.898 227.758
11880   241.358 227.758
12000   241.443 227.758
12120   242.149 227.758
12160   227.758 227.758
12240   227.758 227.758
12360   227.758 227.758
12480   231.391 227.758
12600   231.391 227.758
12720   235.917 227.758
12840   235.917 227.758
12960   237.719 227.758
13080   235.467 227.758
13200   238.106 227.758
13320   238.106 227.758
13371   227.758 227.758
DEPOT - 18 - 21 - 12 - 11 - 8 - 13 - 24 - 14 - 15 - 6 - 7 - 9 - 5 - 4 - 3 - 2 - 23 - 1 - 10 - 16 - 17 - 20 - 22 - 19 - DEPOT
Total distance (cost) = 227.758
Total iteration = 13371
Total computation time = Less than a second
```

Figure 5 Coding of the Tabu Search

The result of this experiment has shown that the best solution for TSP is $Depot - 18 - 21 - 12 - 11 - 8 - 13 - 24 - 14 - 15 - 6 - 7 - 9 - 5 - 4 - 3 - 2 - 23 - 1 - 10 - 16 - 17 - 20 - 22 - 19 - Depot$ with $227.758km$. This solution is generated within less than a second.

Conclusion

This research focused on implementing Tabu Search method to solve the Travelling Salesman Problem. The transport represents the salesman while the attraction places represent the cities. This study was undertaken to design a C++ program and evaluate the given TSP where TS algorithm is applied. As the attracting places in Terengganu and Kelantan arise, the difficulty to solve TSP also increase. As a result, TS was created to assist in the resolution of issues such as TSP, job scheduling problem and others. For optimization problems, TS approach has been shown to yield a global or substantially optimum solution. From the TS method, the optimum route determined was $Depot - 18 - 21 - 12 - 11 - 8 - 13 - 24 - 14 - 15 - 6 - 7 - 9 - 5 - 4 - 3 - 2 - 23 - 1 - 10 - 16 - 17 - 20 - 22 - 19 - Depot$ with the total distance $227.758km$.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Mr. Wan Rohaizad Bin Wan Ibrahim for his time, patience, and encouragement throughout the thesis. Thank you to my parents, Mr Zarimi Bin Awang and Mrs. Norzaimah Binti Abd Wahid for the full support and motivation in completing the task. And thanks to all my friends for their willingness to assist and give motivation when I feel helpless.

References

[1] Basu, S. (2012). Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey. *American Journal of Operations Research*, *02*(02), 163–173. https://doi.org/10.4236/ajor.2012.22019
[2] Chen, J. (2021, December 22). *Heuristics*. Investopedia. https://www.investopedia.com/terms/h/heuristics.asp
[3] Glover, F. (1990). Tabu Search: A Tutorial. *Interfaces*, *20*(4), 74–94. https://doi.org/10.1287/inte.20.4.74
[4] Ma, S. (2020, January 2). *Solving the Travelling Salesman Problem for deliveries*. Routific. https://blog.routific.com/travelling-salesman-problem
[5] Manzano, P. (2021, March 18). *What Is The Vehicle Routing Problem (VRP)?* Routific. https://blog.routific.com/what-is-the-vehicle-routing-problem
[6] Mulder, P. (2022, March 4). *Heuristic Method*. Toolshero. https://www.toolshero.com/problem-solving/heuristic-method/
[7] Ohri, A. (2021, March 11). *Heuristic Method: All You Need To Know In 4 Easy Points*. Jigsaw Academy. https://www.jigsawacademy.com/blogs/cyber-security/heuristic-method/amp/
[8] Osaba, E., Yang, X. S., & del Ser, J. (2020). Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. *Nature-Inspired Computation and Swarm Intelligence*, 135–164. https://doi.org/10.1016/b978-0-12-819714-1.00020-8
[9] Pirim, H., Bayraktar, E., & Eksioglu, B. (2008). Tabu Search: A Comparative Study. *Tabu Search*. https://doi.org/10.5772/5637
[10] Rashid Khan, M. M. U., & Asadujjaman, M. (2016). A Tabu Search Approximation for finding the Shortest distance using Traveling Salesman Problem. *IOSR Journal of Mathematics*, *12*(05), 80–84. https://doi.org/10.9790/5728-1205058084
[11] Uwaisy, M. A., Baizal, Z., & Reditya, M. Y. (2019). Recommendation of Scheduling Tourism Routes using Tabu Search Method (Case Study Bandung). *Procedia Computer Science*, *157*, 150–159. https://doi.org/10.1016/j.procs.2019.08.152
[12] Venkateswarlu, C. (2021). A Metaheuristic Tabu Search Optimization Algorithm: Applications to Chemical and Environmental Processes. *Optimization Problems in Engineering [Working Title]*. https://doi.org/10.5772/intechopen.98240
[13] Wang, Z., Geng, X., & Shao, Z. (2009). An Effective Simulated Annealing Algorithm for Solving the Traveling Salesman Problem. *Journal of Computational and Theoretical Nanoscience*, *6*(7), 1680–1686. https://doi.org/10.1166/jctn.2009.1230
[14] Wikipedia contributors. (2022a, March 27). *Simulated annealing*. Wikipedia. https://en.m.wikipedia.org/wiki/Simulated_annealing
[15] Wikipedia contributors. (2022b, May 5). *Travelling salesman problem*. Wikipedia. https://en.m.wikipedia.org/wiki/Travelling_salesman_problem
[16] Wikipedia contributors. (2022c, May 7). *Vehicle routing problem*. Wikipedia. https://en.m.wikipedia.org/wiki/Vehicle_routing_problem#:%7E:text=The%20vehicle%20routing%20problem%20(VRP,given%20set%20of%20customers%3F%22.&text=The%20objective%20of%20the%20VRP%20is%20to%20minimize%20the%20total%20route%20cost.
[17] Wikipedia contributors. (2022d, May 9). *Heuristic*. Wikipedia. https://en.m.wikipedia.org/wiki/Heuristic
[18] Zhan, S. H., Lin, J., Zhang, Z. J., & Zhong, Y. W. (2016). List-Based Simulated Annealing Algorithm for Traveling Salesman Problem. *Computational Intelligence and Neuroscience*, *2016*, 1–12. https://doi.org/10.1155/2016/1712630