Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 18, 2023, page 155 – 164

Solving Distribution of Petrol Among Petrol Stations Based on Travelling Salesman Problem Using Tabu Search

Tan Zhen Hong, Wan Rohaizad Wan Ibrahim*
Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia
*Corresponding author: wrohaizad@utm.my

Abstract
The purpose of this study is to model the delivery of petrol among petrol stations in Johor Bahru into the Travelling Salesman Problem (TSP) and then solve using the method of Tabu Search (TS). There are several petrol stations selected and the delivery will depart from the company selected and then go back to the departure after visited all the petrol stations. TSP is a well-known routing problem in the world that consider a salesman who must visit some large number of the cities with the objective that minimizing the total distance travelled. This problem also attempts to find the shortest route for the salesman or the vehicle involved to visit all the destinations and then return to the starting point. In addition, the optimal route obtained should not have any interception exists and each destination is only be visited once to minimize the cost. Tabu Search (TS) is the method used in this study to solve the problem and provide the optimal route that satisfy all the objectives proposed. There is an implemented software named Microsoft Visual Studio that supported with the programming language of C++ be used for this study to ease the calculations and save the time since the calculations will involve a large number of calculations and it is definitely an extremely time-consuming process. The result obtained will be the best route that suitably for the delivery of petrol among petrol stations in Johor Bahru that able to minimize the cost and distance travelled.

Keywords Travelling Salesman Problem (TSP); Tabu Search (TS); Optimization

1. Introduction

It is very important to plan a good routing when people want to deliver something from one place to several destinations. No matter delivery company or company in logistic field, it is important to plan the best routing to schedule the job and task of delivery in the best way. There are a lot of advantages that the company can gain if the routing of delivery is followed a good plan. For instance, the company can save the time to delivery, save the cost for the fuel or even maximize the number of products that can be delivered at the same time.

Nowadays, many companies faced the problem on determination of the best route for their delivery service. Therefore, they delivery service might be late from the due date that already set by the customer and it probably will affect the performance of the company. Thus, some company also faced the financial problems as the number of customers had decreased and they need to take action immediately to improve their skills on the determination of the best routing for their delivery service. Then, they can also reduce the cost and achieve the optimal daily operations in a correct way.

This research will focus on the delivery of the petrol among petrol station in Johor Bahru. The delivery of petrol is not that easy because it usually delivered using trucks and there are some limitations

that need to be considered. For example, the drivers of the trucks are illegal to drive in a fast speed as they need to consider the safety of their own and others on road.

2. Literature Review

2.1 Travelling Salesman Problem (TSP)

The Traveling Salesman Problem was first formulated in 1930 by Merrill M. Flood, who looked to solve a school bus routing problem (Keshavdas, 2022). It is a classic algorithmic problem in the field of computer science and operational research. It will focus on optimization, where the best solution is the best route that can minimize the cost or the distance travelled. It is also a well-known mathematical problem that normally can be represent in the form of graph that describing the locations of a set of nodes.

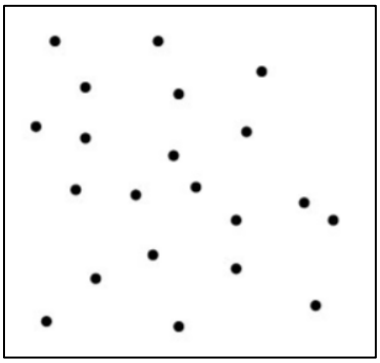


Figure 1 Example of set of nodes

The TSP presents the tasks of finding an optimum path through a set of given locations, such that each location will only be travelled once and the salesman will go back to the starting location after visit all the destinations. In solving this Travelling Salesman Problem (TSP), many researchers will use the heuristics, which provides the possibility outcomes but the result might not be the optimal solution.

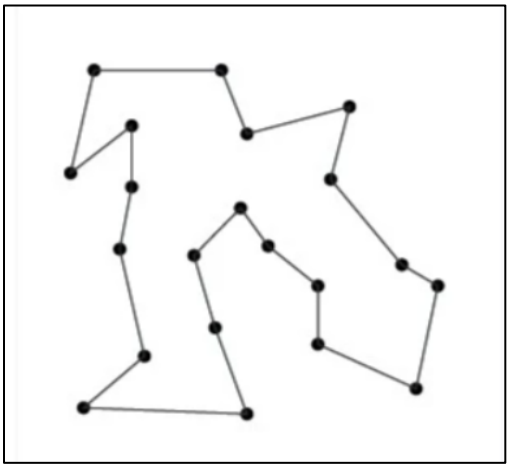


Figure 2 Example of routes connecting all nodes

There are many methods that can be used to solve the Travelling Salesman Problem (TSP). For instance, we can use Simulated Annealing, Tabu Search, Ant Colony Algorithm or many more. However, the method used might need the help of computer programming to calculate the result as it is very difficult and time consuming to solve the problem manually.

2.2 Tabu Search (TS)

Tabu Search is a commonly used meta-heuristic to obtain optimal solution, where metaheuristic optimization deals with optimization problems using metaheuristic algorithms. The concept of optimization arises across a variety of fields, including engineering design, economics, and Internet routing. As money, resources and time are always limited, the optimal utility of these available resources is crucially important. (Yang, 2011). It is often considered that Tabu Search integrates memory structures into local search. Tabu Search is designed to combat a number of issues that can arise as a result of local search, since local search has a wide range of limitations.

The basic idea of Tabu Search is to penalize moves that take the solution into previously visited search spaces (also known as tabu). Tabu Search, however, does deterministically accept non-improving solutions in order to prevent getting stuck in local minimums (Liang, 2020).

2.2.1 Short-term and Long-term Memory

Short Term memory is based off of recency of occurrence and is used to prevent the search algorithm from revisiting previously visited solutions and also can be used to return to good components in order to localize and intensify a search. This is accomplished by the Tabu List and is also known as intensification.

Long Term memory is based off of frequency of occurrence and is used to diversity the search and explore unvisited areas of the search space by avoiding explored areas. This is accomplished by frequency memory and is also known as diversification.

2.2.2 Move

Tabu search moves from one solution to another, been an improvement heuristic in search of a more promising solution. The technique of transitioning from one solution to another is predefined by a set of rules which is known as a move. The neighborhood of the current solution is the series of whole solutions that can be achieved from the solution using a pre-specified move.

2.2.3 Tabu List

To prevent revisiting already seen solutions, Tabu Search employs a tabu list in which tabu moves or characteristics are listed. Moreover, the word tabu is coined from this list of prohibited moves. Tabu lists with short length may not stop cycling outcomes in information loss while on other hand, tabu lists with long length may overmuch expand neighborhood so that moves are fixed to some reach. In essence, if the tabu list is too short it leads to deteriorating the search results. In contrast, if the size of tabu list is too long this means it cannot effectively prevent cycling. Intensification of the search used to decrease the tabu list size whereas diversification tries to increase the size of the tabu list and penalize the frequent move.

2.2.4 Aspiration Criteria

Tabu constraints are subject to an important omission. In a situation where a tabu move has a sufficiently better assessment where it can be evaluated to a solution better than any seen yet, then its tabu categorization may be overruled. Aspiration criterion is the rule that allows such an exception to exist. Aspiration criterion which is frequently used is reverting to a solution better than the last found solution so far.

3. Research Methodology

3.1 Travelling Salesman Problem (TSP)

The objective of Travelling Salesman Problem (TSP) is to visit all the nodes and return back to the starting point with minimization of total distance travelled.
The objective function is formulated as:

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

$$\sum_{i} x_{ij} = 1, \forall\, i \neq j, \sum_{j} x_{ij} = 1, \forall\, j \neq i$$

where

$$x_{ij} = \begin{cases} 1, & Travel\ from\ route\ i\ to\ j \\ 0, & Otherwise \end{cases}$$

3.2 Tabu Search (TS)

3.2.1 Algorithm

Step 1:
We first start with an initial solution $s = s_0$. This can be any solution that fits the criteria for an acceptable solution.

Step 2:
Generate a set of neighboring solutions to the current solution s labeled $N(s)$. From this set of solutions, the solutions that are in the Tabu List are removed with the exception of the solutions that fit the Aspiration Criteria. This new set of results is the new $N(s)$.

$$s' \in N(s) = \{N(s) - T(s)\} + A(s)$$

Step 3:
Choose the best solution out of $N(s)$ and label this new solution s'. If the solution s' is better than the current best solution, update the current best solution. After, regardless if s' is better than s, we update s to be s'.

Step 4:
Update the Tabu List $T(s)$ by removing all moves that are expired past the Tabu Tenure and add the new move s' to the Tabu List. Additionally, update the set of solutions that fit the Aspiration Criteria $A(s)$. If frequency memory is used, then also increment the frequency memory counter with the new solution.

Step 5:
If the Termination Criteria are met, then the search stops or else it will move on to the next iteration. Termination Criteria is dependent upon the problem at hand but some possible examples are:
i) a maximum number of iterations
ii) if the best solution found is better than some threshold

3.2.2 Stopping Criterion

Some prompt stopping conditions are:
i) A given number of fixed iterations.
ii) A given amount of processor time.
iii) No feasible moves into the locality of the current found solution.
iv) Evidence can be given that an objective functions output is feasible.

4. Software Implementation

4.1 Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) that used to develop computer program. This software supports some programming languages, mainly for C and C++.

The use of this software is to speed up the computation of using Tabu Search since the computation might involve more than 10000 iterations and there is impossible to calculate manually in a short period.

4.1.1 Interface

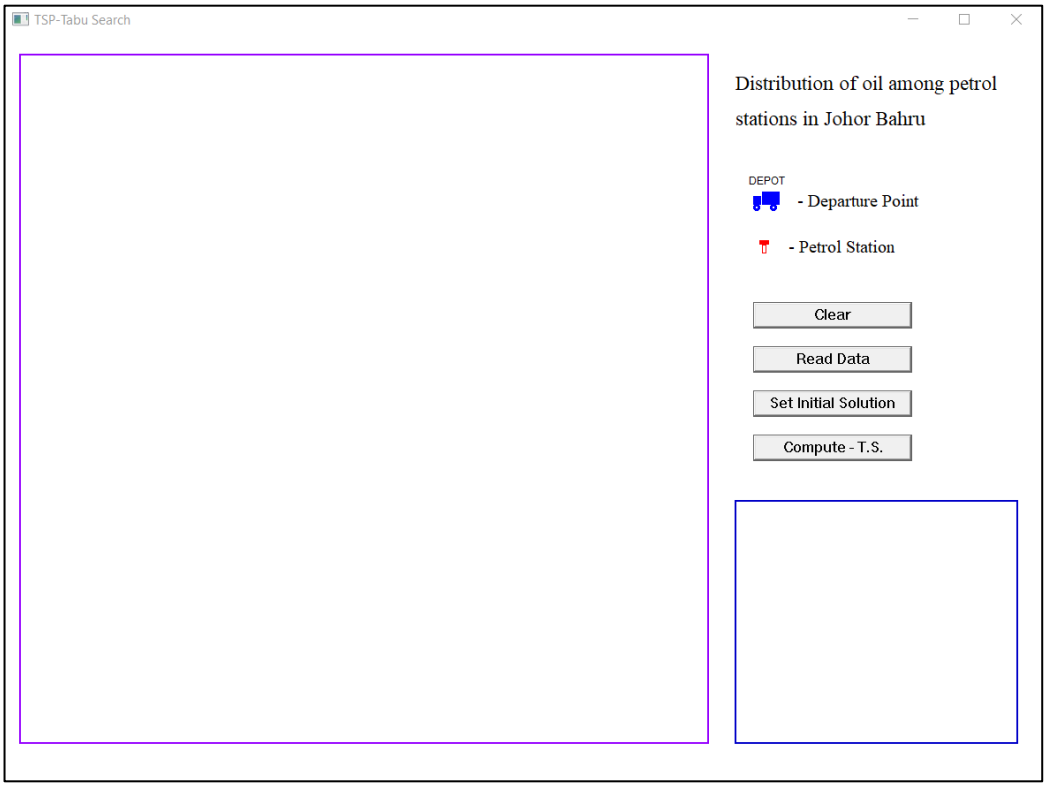


Figure 3 Interface created

Based on the figure above, the interface created considered a box to plot the locations of the petrol stations and the company selected. Besides, there is also another box that will list down the result obtained which are the total number of clients (petrol stations), initial solution, solution using Tabu Search, number of iterations involved in the calculation and the time for computing the solution.

Moreover, there are several buttons that created to be used for the calculations. Once the button be clicked, then they will link to the specific actions that be assigned.

i) Clear
Clear screen and all the variables involved, reset the calculation

ii) Read data
Read the data from input file and plot the data into the box

iii) Set initial solution
Calculate and set an initial solution, draw the initial route

iv) Compute-T.S.
Compute the solution using Tabu Search

4.1.2 Input File

The C++ Programming will read the input data from an input file. The input file is named as “input.txt”. The coordinates of the locations of petrol stations and the company selected is listed in an input file. The correct way to list down the coordinates is the coordinate x will be listed first, then leave a space and then followed by the coordinate y. The listed locations in input file will start from the company which placed at the top and the program will automatically consider it as a departure point, then followed by the 25 petrol stations selected.

4.1.3 Output File

The C++ Programming will also provide the output in an output file. For this case, there are two output files be assigned. The first output file is named as “initialout.txt”, which contained the result of the initial solutions. Another output file which named as “outTS.txt” listed the result computed using Tabu Search. Both the output files provided the cost, number of iterations and the computing time.

5. Result and Discussion

5.1 Data

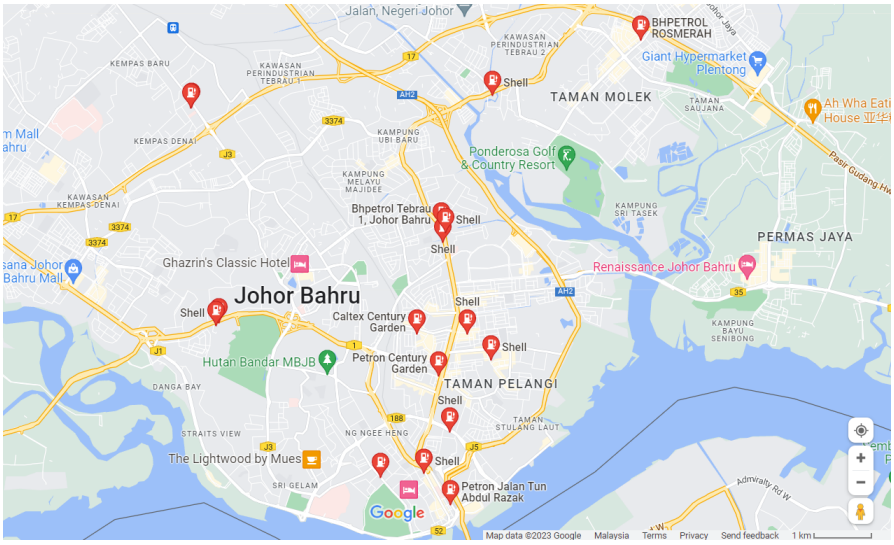


Figure 4 Locations of the petrol stations in Johor Bahru

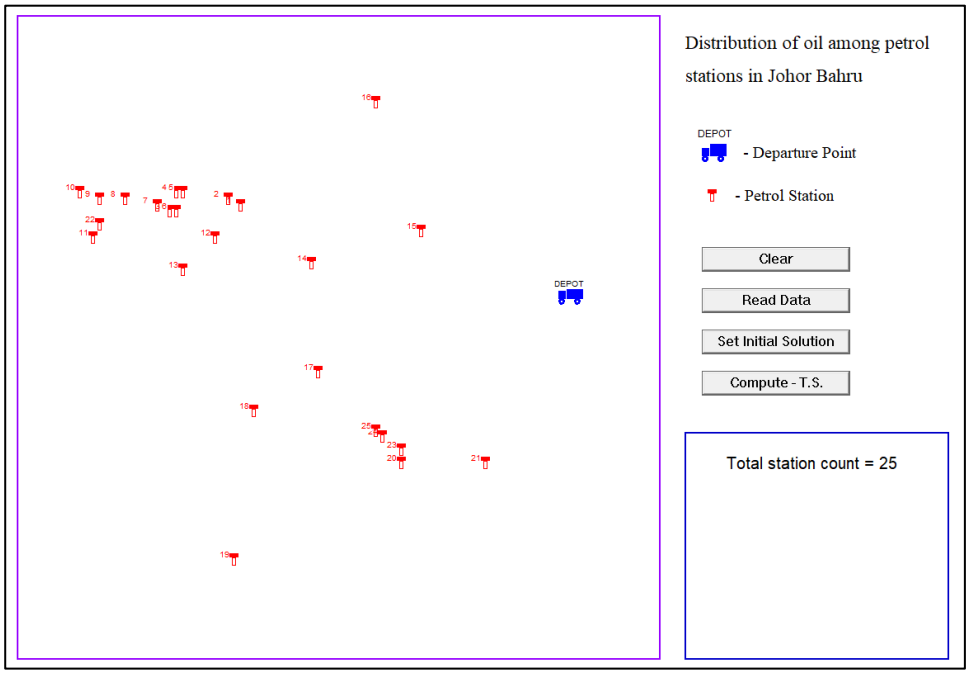
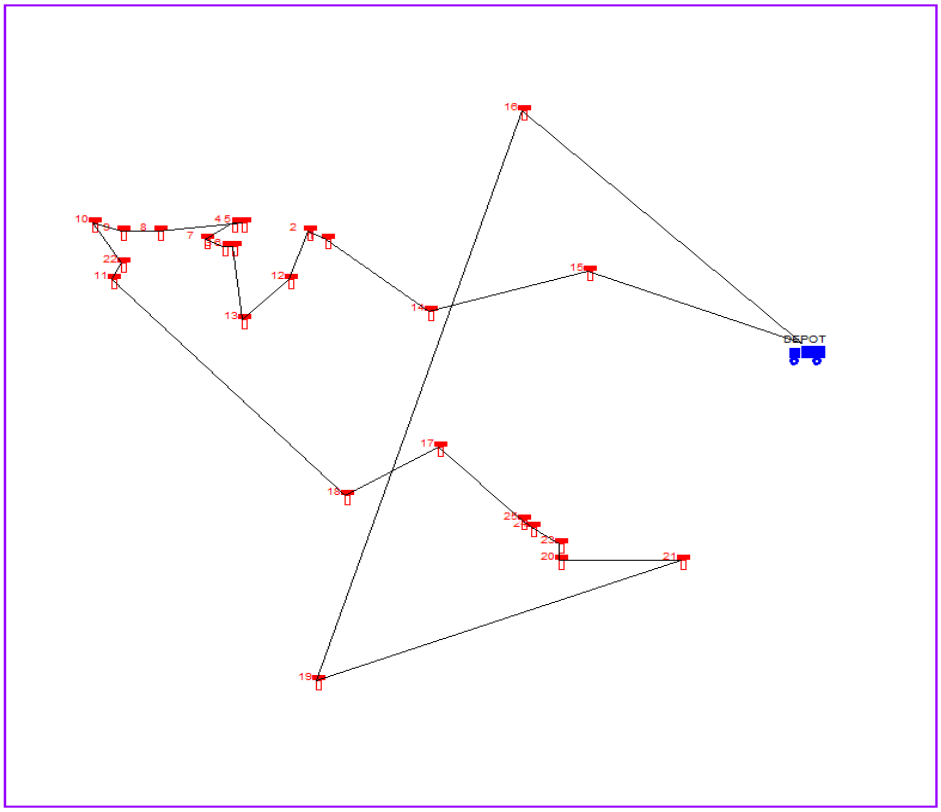


Figure 5 Plotted of locations in Microsoft Visual Studio

5.2 Initial Solution




```
DEPOT - 15 - 14 - 1 - 2 - 12 - 13 - 6 - 3 - 7 - 4 - 5 - 8 - 9 - 10
- 22 - 11 - 18 - 17 - 25 - 24 - 23 - 20 - 21 - 19 - 16 - DEPOT

Initial Cost = 352.491
```

Figure 6 Initial Solution obtained

5.3 Computation using Tabu Search

5.3.1 Single Run

Table 1: Result obtained for Single Run

Run	Number of iterations	Optimal Cost	Route
Initial Run	-	352.491	DEPOT - 15 - 14 - 1 - 2 - 12 - 13 - 6 - 3 - 7 - 4 - 5 - 8 - 9 - 10 - 22 - 11 - 18 - 17 - 25 - 24 - 23 - 20 - 21 - 19 - 16 – DEPOT
1st Run	14370	290.155	DEPOT - 15 - 16 - 14 - 1 - 2 - 5 - 4 - 7 - 8 - 9 - 10 - 11 - 22 - 3 - 6 - 12 - 13 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 – DEPOT
2nd Run	14335	281.989	DEPOT - 15 - 16 - 1 - 2 - 4 - 5 - 3 - 6 - 7 - 8 - 10 - 9 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT
3rd Run	14312	**278.226**	**DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT**
4th Run	14382	284.649	DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 8 - 9 - 10 - 22 - 11 - 7 - 6 - 3 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT
5th Run	14326	284.307	DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 10 - 9 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 20 - 25 - 24 - 23 - 21 - DEPOT
6th Run	14248	279.315	DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 3 - 6 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT
7th Run	14327	**278.226**	**DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT**
8th Run	14355	281.433	DEPOT - 15 - 16 - 2 - 1 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT
9th Run	14326	281.433	DEPOT - 15 - 16 - 2 - 1 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT
10th Run	14316	287.519	DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 8 - 9 - 10 - 11 - 22 - 7 - 3 - 6 - 12 - 13 - 14 - 17 - 18 - 19 - 25 - 24 - 20 - 23 - 21 - DEPOT
Best Solution	**2 out of 10 (20%)**	**278.226**	**DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 - DEPOT**

Based on the table above, the best solution obtained is 278.226, which is much smaller compared to the initial solution of 352.491. There is 2 times out of 10 runs that are able to produce the best solution that achieve the objective which seek for the minimum cost. The best route obtained will be followed DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 – DEPOT, which only visit each petrol station once and there do not exist any interception between the route and the delivery will end up at the departure point after visited all the petrol stations.

5.3.2 Multiple Run

Table 2: Result obtained for Multiple Run

Run	First trial		Second trial		Third trial	
	Number of iterations	Optimal Cost	Number of iterations	Optimal Cost	Number of iterations	Optimal Cost
Initial Run		352.491		352.491		352.491
1st Run	14339	282.873	14237	282.140	14307	283.512
2nd Run	14268	279.380	14252	279.315	14296	**278.226**
3rd Run	14243	**278.226**	14294	**278.226**	14293	278.226
Best Solution		**278.226**		**278.226**		**278.226**

Based on the table above, the best solution will be obtained among the first three runs for the 3 trials. Besides, the number of iterations for obtaining the best solution by using the optimal solution from previous run as the initial solution had been decreased.

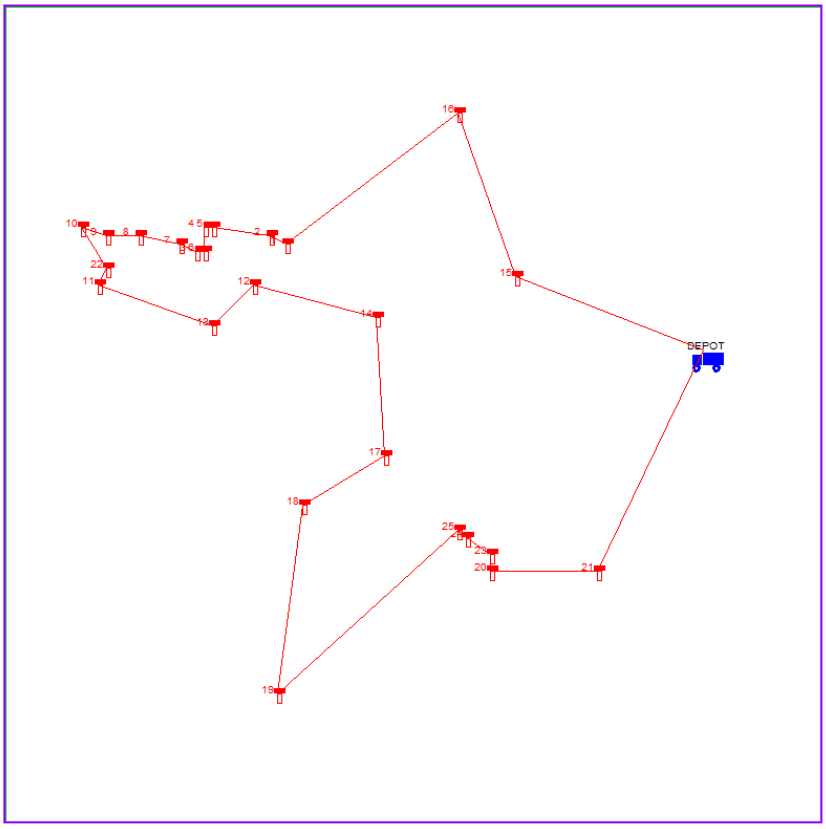


Figure 7 Best solution obtained

6. Conclusion

In this study, the best solution is obtained by using the Tabu Search and the route do not contain any interception that will waste the time and money for the delivery. The best solution obtained is 278.226. The result obtained is much better compared to the initial solution calculated, which is 352.491. The

route suggested will be followed the sequence of DEPOT - 15 - 16 - 1 - 2 - 5 - 4 - 6 - 3 - 7 - 8 - 9 - 10 - 22 - 11 - 13 - 12 - 14 - 17 - 18 - 19 - 25 - 24 - 23 - 20 - 21 – DEPOT, which will give the best solution that satisfy all the objectives proposed.

Based on the result obtained in the case of Single Run, the optimal cost and route obtained for each run showed that a small difference in the choosing of the path will lead to a significant difference in the cost. There are some run that came out with a very similar route, but the optimal cost obtained is different. Besides, the result obtained for the case of Multiple Run showed that the decision on the initial solution will improve the speed of the computation by reducing the number of iterations.

References

[1] Arora, S. (1998). Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. Journal of the ACM (JACM), 45(5):753–782.

[2] Czumaj, A. (2016). Euclidean Traveling Salesman Problem. In: Kao, MY. (eds) Encyclopedia of Algorithms. Springer, New York, NY. https://doi.org/10.1007/978-1-4939-2864-4_131

[3] Drabowski, M. (2019). Short-Term and Long-Term Memory in Tabu Search Algorithm for CAD of Complex Systems with Higher Degree of Dependability. DepCoS-RELCOMEX 2018. Advances in Intelligent Systems and Computing, vol 761. Springer, Cham. https://doi.org/10.1007/978-3-319-91446-6_17

[4] Fred, G. (1990). Tabu Search: A Tutorial. Interfaces 20(4):74-94. https://doi.org/10.1287/inte.20.4.74

[5] Keshavdas, M. (2022). What Is The Traveling Salesman Problem? Fleetroot. https://fleetroot.com/blog/what-is-the-traveling-salesman-problem/

[6] Kodeeswaran. (2022). Travelling Salesman Problem (TSP). InterviewBit. https://www.interviewbit.com/blog/travelling-salesman-problem/

[7] Kuo, M. (2020). Solving The Travelling Salesman Problem For Deliveries. Routific Blog. https://blog.routific.com/blog/travelling-salesman-problem

[8] Latitude and Longitude in Decimal Degrees. Surfer Help. https://surferhelp.goldensoftware.com/projections/wks_lat_long_in_decimal_degrees.htm

[9] Liang, F. (2020). Optimization Techniques — Tabu Search. Towards Data Science. https://towardsdatascience.com/optimization-techniques-tabu-search-36f197ef8e25

[10] Rahman, R. A., Mokhtar, N. E., & Bahrom, U. L. (2017). Solving A Fuel Distribution Problem Using Genetic Algorithm: A Traveling Salesman Problem Approach. Journal of Technology and Operations Management, 12(1), 51–55. https://doi.org/10.32890/jtom2017.12.1.6.

[11] Roberta, H and Alison, T. (2015). Validity and reliability in quantitative studies. BMJ Journals. https://ebn.bmj.com/content/18/3/66

[12] Subham, D. (2022). Traveling Salesman Problem – Dynamic Programming Approach. Baeldung. https://www.baeldung.com/cs/tsp-dynamic-programming

[13] Understanding Latitude and Longitude. Journey North. https://journeynorth.org/tm/LongitudeIntro.html

[14] Yang, X. S. (2011). Metaheuristic Optimization. Scholarpedia, 6(8):11472. http://www.scholarpedia.org/article/Metaheuristic_Optimization