Review of Algorithms to Solve Travelling Salesman Problem

OGBULOKO VINCENT ECHE[1], ADEREMI ELISHA OKEYINKA[2], IBRAHIM ABDULLAH[3], ABDULGANIYU ABDULRAHMAN[4]
[1, 2, 3, 4]*Department of Computer Science, Faculty of Natural Sciences, Ibrahim Badamasi Babangida University, Lapai, Niger State, Nigeria.*

Abstract - The Travelling Salesman Problem (TSP) is a popular optimization problem in which shortest path of the salesperson travelling to all cities once and returning to the origin city is to be determined. This is done either by using exact algorithms or heuristic algorithms. The main concern with exact algorithms is that; exact algorithms can produce optimal solution but are always not practicable due to complexity of combinatorial optimization problem which are mostly NP- hard and the constraint of time. Therefore, TSP is solved using various heuristic algorithms which produce good enough but not necessarily optimal solution in reasonable time and drastically cuts down the solution space. This paper presents a review of different algorithms to solve TSP and find the shortest path through all the cities that the salesperson has to travel.

Indexed Terms- Combinatorial Optimization, Exact Algorithm, Heuristic Algorithm, TSP, NNH, NIH, FIH, CIH, RIH, NP – Hard.

1. INTRODUCTION

The travelling salesman problem is an optimization problem which involves finding the shortest path to visit every city and return to the starting point. It is one the widely studied problems in combinatorial optimization. An optimal path has to be determined in terms of time and cost value through a number of cities which is to be travelled by the salesperson to reach his destination in such a manner that he travels each city exactly once and returns to the origin city. The TSP is NP – hard because no known algorithm exists to solve it in polynomial time.

Many researchers have tried to solve TSP using different algorithms such as branch & bound, dynamic programming, Greedy algorithm, artificial bee colony (ABC), Local search algorithm, ant colony optimization (ACO), Simulated annealing (SA), Hill – climbing algorithm, Genetic algorithm (GA), Covering path planning algorithm (CPP), Cuckoo search algorithm (CS), etc.

This paper gives the review of different algorithms used to solve the travelling salesman problem and a detailed description of the most recently introduced hybridized Full Edge Search algorithms (hybridized FES).

II. CLASSICAL ALGORITHM TO SOLVE TRAVELLING SALESMAN PROBLEM (TSP)

Different exact algorithms such as Dynamic Programming and Linear Programming have been used to solve the travelling salesman problem.

1.1 Dynamic Programming (DP)

Dynamic programming is a commonly used algorithm technique for efficiently computing recurrences by storing partial results and reusing them when needed. It is a method for solving complex or difficult problem by breaking it down into sub-problems. It involves simple formulation of the approach and critical thinking and programming. The idea is simply, if you have solved a problem with the given input, then save the result for future reference, so as to avoid solving the same problem again; simply remember your past. If the given problem can be broken up into smaller sub - problems and these sub - problems can still be broken into smaller ones and in the process, you observe overlapping sub - problems, then it is a pointer for Dynamic Programming.

Dynamic Programming can be achieved using two approaches;

i. Top - Down approach: Start solving the given problem by breaking it down. If you observe that the problem has been solved, just returns the saved answer. If the problem has not been solved, solve it and save the answer. This is easy to think and very intuitive. It is known as Memoization
ii. Bottom - up approach: Observe the problem and look at the order in which the sub - problems are solved and start solving from the simpler sub - problem up towards to the final solution. This process is called Tabulation.

When implementing dynamic programming, the following steps are useful;

i. Define a class of sub - problems
ii. Create a recurrence based on solving each sub - problem using simpler sub - problems
iii. Create an algorithm for computing the recurrence
iv. Add Memoization to save solutions to sub - problems to avoid repetition of solution to sub - problem.

This approach is also used to solve travelling salesman problem but only for limited number of cities.

III. Tour Construction Heuristic Algorithms to solve the Travelling Salesman Problem

Different tour construction optimization algorithms such as Nearest Neighbor Algorithm (NN), Nearest Insertion Algorithm (NI), Farthest Insertion Algorithm (FI), Cheapest Insertion Algorithm (CI) and Random Insertion Algorithm (RI) have been used to solve the travelling salesman problem. These algorithms will be described in this section

3.1 Nearest Neighbor Algorithm

The nearest neighbor algorithm was one of the algorithms used to solve the travelling salesman problem approximately. It is a simple and intuitive approximation for the TSP. This is a greedy approach. The greedy criterion is selecting the nearest city. It starts at an arbitrary city and repeatedly selects the nearest unvisited city until all cities have been visited.

The algorithm steps are as follows:
Step i: Start at any city
Step ii: Select the nearest unvisited neighbor and add it to end of tour.
Step iii: Repeat step (ii) until all nodes are added to the tour.

Time complexity = $O(n^2)$

3.2 Nearest Insertion Algorithm

Nearest Insertion Algorithm is still greedy but not as greedy as nearest neighbor. It allows partial tour to be modified. The algorithm start with a sub - tour of one city, then finds the city that is nearest to it and inserts it into the sub - tour. The algorithm continue to find the nearest city and insert until all city are added to the tour

The algorithm steps are as follows:
Step I: Start the tour at any city
Step ii: Pick the nearest unvisited - neighbor of the selected city in the tour
Step iii: Insert it into the tour $T = t_1, \ldots, t_k$ so that the total tour distance (cost) is minimized. i.e., find (i,j,k) = w(i, k) + w(k, j) - w(i, j) is minimize
Step iv. Repeat steps (ii) and (iii) until all cities are added to the tour.

Time complexity = $O(n^2)$

3.3 Farthest Insertion Algorithm

The farthest insertion algorithm is a heuristic used to solve the travelling salesman problem. It start with a random city and repeatedly finds the city farthest from any city in the tour, then inserts it into the tour in a way that minimizes the increase in the tour distance.

The algorithm steps are as follows:
Step I: Start the tour at any city
Step ii: Pick the nearest farthest unvisited neighbor of the selected node.
Step iii: Insert it into the tour $T = t_1, \ldots, t_k$ so that the total tour distance (cost) is minimized. i.e., find (i,j,k) = w(i, k) + w(k, j) - w(i, j) is minimized

Step iv: Repeat steps (ii) and (iii) until all nodes are added to the tour.
Time complexity = $O(n^2)$

3.4 Cheapest Insertion Algorithm

The cheapest insertion algorithm is heuristic that works by building a tour from small cycles with minimal weight and then adding new cities. It selects the new city and the weight simultaneously to obtain the minimum insertion distance.

The algorithm steps are as follows:

Step i: Start with a partial tour from a city

Step ii: Create a sub tour relationship; a sub tour link is created between two (2) places. It is a journey from the first place and ends in the first place.

Step iii: Change the direction of the relationship (insertion) . One of the directions of the relationship (arc) of two places with a combination of two arcs, namely arc (i,j) is change to arc (i,k) and arc (k,j) where k is the insertion point with the smallest additional distance which is obtained from (i,j,k) = w(i,k) + w(k,j) - w(i,j).

Step iv: Repeat steps (ii) and (iii) until all nodes are added to the tour.

Time complexity = $O(n^2 \log n)$

3.5 Random Insertion Algorithm

The random insertion algorithm is a heuristic method for solving the travelling salesman problem. It start by selecting a city randomly and the nest city to join the tour is also randomly selected and inserted into the tour where it causes minimal increase in the tour length. The procedure is repeated until all cities are inserted into the tour.

The algorithm steps are as follows:

Step i: Select a random city (or a pre-specified city) as the initial tour.

Step ii: Randomly the next city to join the tour from the remaining cities (not yet connected to the tour).

Step iii: Calculate the cost of inserting a city between two existing cities in the tour. The cost is defined as: (i,j,k) = w(i, k) + w(k, j) - w(i, j) is minimize. Insert the selected city at the location that minimizes the insertion cost.

Step iv. Repeat steps (ii) and (iii) until all cities are added to the tour.

Time complexity = $O(n^2)$.

3.6 Hybridized Full Edge Search Algorithm

The Hybrid Full Edge Search (Hybrid FES) algorithm is a new heuristic that solves the travelling salesman problem approximately. It starts by sorting all the edges of n cities and selects the city with the shortest edges as the origin city. It selecst the next nearest city and inserts it into the tour in the point where it causes the minimal increase in the tour length.

The algorithm steps are as follows:

Step i: Sort all edges from n nodes

Step ii: Select the edge with a short distance

Step iii: The tour begins with a city with the shortest possible edge

Step iv: Mark the city as visited

Step v: Search for the next closest city

Step vi: Insert the closest city into the tour so that the total tour distance is minimized. Insertion point into the tour is define as; (i, k, j) = w(i, k) + w(k, j) – w(i, j)

Step vii: Is there any city not yet visited? If yes, GOTO step 5

Step viii: Return to the first chosen edge's starting node

CONCLUSION

The travelling salesman problem is one of most important combinatorial optimization problem in graph theory. The different algorithms used by researchers to find solution to the travelling salesman problem have been reviewed and presented in this paper. It is found that the Hybrid FES algorithm contains the best features of others algorithms which make the algorithm performed better and suitable for large sized optimization problem.

REFERENCES

[1] Asani, et al (2020). A construction tour technique for solving the travelling salesman problem based on convex hull and nearest neighbor

heuristics. *International Conference on Mathematics, Computer Engineering and Computer Science*, 1–4.

[2] Asani, et al (2024). A Novel Insertion Solution for the Travelling Salesman Problem. *Computers, Materials & Continua*.

[3] Babel, (2020). New heuristic algorithms for the Dublin traveling salesman problem. *Journal of Heuristics.*

[4] Bernardino and Paias (2018). Solving the family traveling salesman problem. *European Journal of Operational Research. 267 (2).*

[5] Donald and Magdalena (2020). Traveling Salesman Problem – An Overview. Introductory chapter. *DOI:10.5772/intechopen.94435.*

[6] Fadhillah, et al (2017). Solving travelling salesman problem using heuristics approaches. *Journal of Global Research in Computer Science/*

[7] Farid, et al (2022). Implementation of Cheapest Insertion Heuristic Algorithm in Determining Shortest Delivery Route. *International Journal of Global Research*, *3*(2), 37–45.

[8] Hoffman, et al (2016). Traveling Salesman Problem. *Encyclopedia of Operation Research and Management Science. Pp. 1573 – 1578.*

[9] Krari, et al (2021). A pre – processing reduction method for the generalized travelling salesman problem. *European Journal of Operation Research. 21(11)*

[10] Kumar, et al (2012). A Genetic Algorithm Approach to study travelling salesman problem. *Journal of Global Research in Computer Science. 3(3):33-8*

[11] Laha, et al (2016). Nature – Inspired Metaheuristics for optimizing Information Dissemination in Vehicular Networks. *TECNALIA Research and Innovation, Derio, Spain.*

[12] Lity, et al (2022). Travelling salesman problem. An overview of Application. *International Journal of Advanced Research in Education Technology.*

[13] Malik and Muhammad (2015). Heuristic Approaches to solve travelling salesman problem. *TELKOMNIKA Indonesian Journal of Electrical Engineering. 15(2). Pp. 390-396*

[14] Nemani, et al (2021). Algorithms and Optimization techniques for solving traveling salesman problem

[15] Ono, et al (2020). A criteria – based approach to the travelling salesman problem. *Western Decision Science Journal. Librarian Publications and Presentations. 143.*

[16] Rahman, et al (2024). Improvement of the Nearest Neighbor Heuristic Search Algorithm for Travelling Salesman Problem. *Journal of Engineering Advancements*, *5*(1), 19–26.

[17] Rao, et al (2024). Literature survey on travelling salesman problem using genetic algorithms/ *International Journal of Advanced Research in Education Technology.* 26, pp. 503 – 530.

[18] Sathya and Muthukumaravel (2015). A review of the optimization algorithms on travelling salesman problem. *Indian journal of science and technology. 8(29)*

[19] Sharma and Dutta (2015). Review of Algorithms to solve travelling salesman problem. *Journal of Basic and Applied Engineering Research. 2(18), pp. 1612-1616.*

[20] Sundar and Rathinam (2016). Generalized multiple depots traveling salesman problem. *Computers and Operation Research. 70*

[21] Tawanda, et al (2023). A labelling Method for the Travelling Salesman Problem. *Journals of Applied Sciences. 13(11)/ 103390/app13116417.*

[22] Yuan, et al (2020). A branch and Cut algorithms for the generalized travelling salesman problem. *European Journal of Operational Research. 286(3)*

[23] Zhu, et al (2022). Algorithm for Solving Traveling Salesman Problem Based on Self – Organizing Mapping Network. *Journal of Shanghai Jiaotong University (Science). 29. Pp. 463 – 470*