`

|  |
| --- |
| **DAYANANDA SAGAR UNIVERSITY**  **Devarakaggalahalli, Harohalli, Kanakapura Road, Ramanagara Dt, Karnataka 562112** |

**Bachelor of Technology**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence and Machine Learning)**  


**Mini Project**

**(TRAVELLING SALESMAN PROBLEM)**

By

**Shrinivas Sanjay Joshi-ENG22AM0062**

**Shashwat Dodmani-ENG22AM0058**

**Rajat S U-ENG22AM0045**

**Pruthviraj S R-ENG22AM0042**

**Under the supervision of**

**Prof. Pradeep Kumar K**

**Dr. Mary Jasmine**

**Prof. Mitha Guru**

**Assistant Professor, Artificial Intelligence & Machine Learning, SOE**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**(Artificial Intelligence and Machine Learning)**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY,   
BANGALORE**



**School of Engineering**

**Department of Computer Science & Engineering**

**(Artificial Intelligence and Machine Learning)**

Devarakaggalahalli, Harohalli Kanakapura Road, Ramanagara Dt, Karnataka 562112

**CERTIFICATE**

This is to certify that the Mini – Project titled **“Travelling salesman problem”**

is carried out by **Shrinivas Sanjay Joshi(ENG22AM0062),Shashwat Dodmani(ENG22AM0058),Rajat S U(ENG22AM0045),Pruthviraj S R(ENG22AM0042),** bonafide students of Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence and Machine Learning) at the School of Engineering, Dayananda Sagar University,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof. Pradeep Kumar K Dr. Jayavrinda Vrindavanam V**

**Prof. Mitha Guru** Chairman CSE (AI &ML)

**Dr. Mary Jasmine** School of Engineering

Assistant Associate professor Dayananda Sagar University

Dept of CSE(AI&ML)

School of Engineering

Dayananda Sagar University

Date: Date:

**ACKNOWLEDGEMENT**

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor’s degree in this institution.

We would like to thank **Dr. Udaya Kumar Reddy K R,** **Dean**, **School of Engineering**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrindavanam,** **Department Chairperson**, **Computer Science, and Engineering (Artificial Intelligence and Machine Learning)**, **School of Engineering, Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Prof. Pradeep Kumar K, Prof. Mitha Guru and Dr. Mary Jasmine** **Assistant Professor**, **Dept. of Computer Science and Engineering (Artificial Intelligence and Machine Learning)**, **School of Engineering,** **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our UG Research project work, which paved the way for smooth progress and the fruitful culmination of the research.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Research work.

# TABLE OF CONTENTS

Page  
[LIST OF ABBREVIATIONS ........................................................................................... vi](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250008)

[LIST OF FIGURES ………............................................................................................. vii](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250007)

LIST OF TABLES …………........................................................................................... viii

[ABSTRACT .................................................................................................................... ix](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250006)

[CHAPTER 1 INTRODUCTION………........................................................................ 1](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250005)

1.1 MODEL……………………………………………………………………...

[1.2 GENETIC ALGORITHM…………………………………………………](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250003)... 2

CHAPTER 2 LITERATURE SURVEY. ……….......................................................... 4

[CHAPTER 3 PROBLEM DEFINITION..................................................................](https://d.docs.live.net/e96bfddd435d8bba/Documents/AI_Report%5b1%5d.docx#_TOC_250002)....... 6

CHAPTER 4 METHODOLOGY ……………………………………………………... 7

4.1. OBJECTIVE FUNCTION ............................................................................ 8

4.2. INITIALIZATION ………………………………………………………… 8

4.3 SELECTION ………………………………………………………………. 10

4.4 CROSSOVER………………………………………………………………

4.5 MUTATION ……………………………………………………………….

4.6 EVOLUTION ………………………………………………………………

4.7 VISUALIZATION …………………………………………………………

CHAPTER 5 RESULTS AND ANALYSIS …................................................................ 11

CHAPTER 6 REFERENCES…………………………………………………………… 15

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| TSP | Travelling Salesman Problem |
| AI | Artificial Intelligence |
| OSM | Open Street Map |
| NP | Non-Deterministic Polynomial |
| ILP | Integer Linear Programming |
| GA | Genetic algorithm |
| VLSI | Very Large Scale Integrated |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig No. | Description of the figure | Page No. |
| 1.1 | Travelling Salesman Problem |  |
| 4.1.1 | Mutation of Chromosomes |  |
| 5.1 | Optimal Solution of Travelling Salesman Problem |  |
| 5.2 | Output of the code implementation using Genetic Algorithm |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Fig No. | Description of the figure | Page No. |
| 5.3 | Sample output of the code implementations |  |

.

**ABSTRACT**

Travelling Salesman Problem (TSP) can be applied to find the most efficient route to travel between various nodes. The goal is to make smart cities to be created by heuristic algorithms on the real maps to perform some tasks through TSP. Therefore, Hill Climbing heuristic search algorithm which is generally used for mathematical optimization problems in Artificial Intelligence (AI) field has been preferred in this study. This application takes a city from the OpenStreetMap (OSM), which is a real map as an input given to the algorithm, and calculates a path to visit all the nodes on the related route. The output was intended to be found in the shortest possible way and in the least possible time. On the market, there are some travelling, public transport and discovery applications or games on the smart maps. Also there are some publications about TSP and metaheuristic approaches in the literature but the sources are generally commercial products and for limited cities. There is no application that takes all the cities as a source and makes a travel plan for tourists. This study intended to create an opensource, location independent travel plan advisor and develop an indigenous product. Application was tested for Rome and Ankara as an instance but because a flexible working area OSM was used, application can be generated for all the routes and also various applications can be developed by researchers based on this study.

**CHAPTER 1**

**INTRODUCTION**

The Traveling Salesman Problem (TSP) stands as one of the most renowned combinatorial optimization problems in the realm of computer science and mathematics. It involves finding the shortest possible route that visits a set of cities exactly once and returns to the starting city. Originally formulated in the 1800s, the problem has numerous practical applications, including logistics, planning, manufacturing, and DNA sequencing.

Mathematically, the challenge intensifies as the number of cities increases due to the exponential growth in potential routes. TSP is classified as NP-hard, meaning there's no known algorithm that efficiently solves all instances in polynomial time.

Various algorithms and heuristics have been devised to tackle TSP, ranging from exact algorithms like branch and bound to approximate methods such as genetic algorithms, ant colony optimization, and simulated annealing. The significance of TSP extends beyond theoretical interest, as its solutions have practical implications in optimizing travel routes, circuit board drilling, and delivery logistics, among other real-world scenarios.

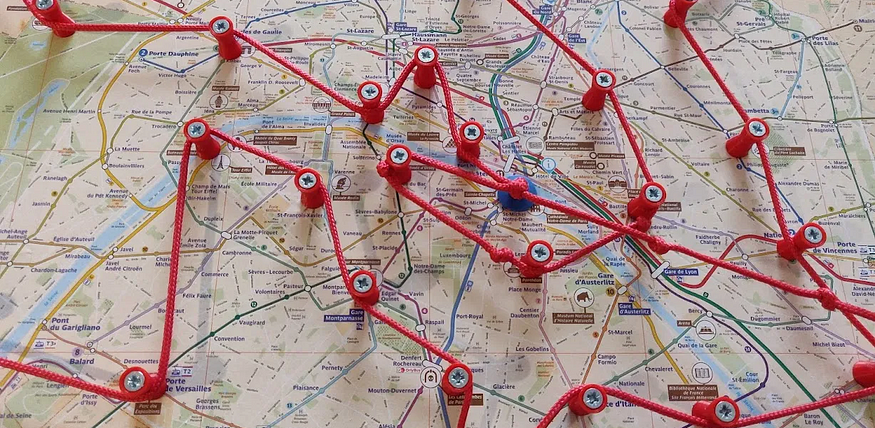


Fig 1.1 Travelling Salesman Problem

* 1. **Model**

Various models can be developed for the Traveling Salesman Problem (TSP). Classical models include Integer Linear Programming (ILP) formulations, where decision variables represent

the order of city visits. Additionally, graph-based models, such as adjacency matrix or adjacency list representations, facilitate algorithmic solutions. Reinforcement learning models, like Q-learning, can be trained to discover optimal or near-optimal TSP solutions through trial and error. Evolutionary algorithms, such as Genetic Algorithms, encode potential tours in chromosomes and e

volve them over generations. Hybrid models combining machine learning and optimization techniques have also gained attention. These models showcase the diverse approaches available to address the TSP, each with its strengths and applicability in different contexts

* 1. **Genetic Algorithm**

Genetic algorithms (GAs) are optimization algorithms inspired by the principles of natural selection and evolution. In the GA framework, potential solutions to a problem are represented as individuals or "chromosomes" in a population. Each chromosome undergoes evaluation through a fitness function, which quantifies its effectiveness in solving the problem. The fittest individuals, those with higher fitness scores, are more likely to be selected for reproduction. During reproduction, pairs of individuals exchange genetic material through crossover, emulating genetic recombination. Additionally, random changes are introduced via mutation, fostering diversity in the population. Over successive generations, the algorithm evolves by favoring individuals with superior traits. The process continues until a termination condition is met, such as reaching a specified number of generations or achieving a satisfactory solution. GAs are applicable to a broad spectrum of optimization problems, offering a robust and adaptive approach to finding near-optimal solutions in complex and dynamic solution spaces.

**CHAPTER 2**

**LITERATURE REVIEW**

# Base Paper: Genetic Algorithm for Travelling Salesman Problem with Modified Cycle Crossover Operator by M. Nauman Sajid, Ijaz Hussain, Alaa Mohamd Shoukry and Showkat Gani

# https://www.hindawi.com/journals/cin/2017/7430125/

Genetic algorithms (GAs) are derivative-free stochastic approach based on biological evolutionary processes proposed by Holland. In nature, the most suitable individuals are likely to survive and mate; therefore, the next generation should be healthier and fitter than previous one. A lot of work and applications have been done about GAs in a frequently cited book by Golberg. GAs work with population of chromosomes that are represented by some underlying parameters set codes.

The traveling salesman problem (TSP) is one of the most famous benchmarks, significant, historic, and very hard combinatorial optimization problem. TSP was documented by Euler in 1759, whose interest was in solving the knight’s tour problem. It is the fundamental problem in the fields of computer science, engineering, operations research, discrete mathematics, graph theory, and so forth. TSP can be described as the minimization of the total distance traveled by touring all cities exactly once and return to depot city. The traveling salesman problems (TSPs) are classified into two groups on the basis of the structure of the distance matrix as symmetric and asymmetric. The TSP is symmetric if , , where  and  represent the row and column of a distance (cost) matrix, respectively, otherwise asymmetric. For  cities, there are  possible ways to find the tour after fixing the starting city for asymmetric and its half for symmetric TSP. If we have only 10 cities, then there are 362,880 and 181,440 ways for asymmetric and symmetric TSP, respectively. This is the reason to say TSP is NP-hard problem. TSP has many applications such as variety of routing and scheduling problems, computer wiring, and movement of people, X-ray crystallography, and automatic drilling of printed circuit boards and threading of scan cells in a testable Very-Large-Scale-Integrated (VLSI) circuits.

Over the last three decades, TSP received considerable attention and various approaches are proposed to solve the problem, such as branch and bound, cutting planes, particle swarm, simulated annealing, ant colony, neural network, tabu search, and genetic algorithms. Some of

these methods are exact, while others are heuristic algorithms. A comprehensive study about GAs approaches are successfully applied to the TSP. A survey of GAs approaches for TSP was presented by Potvin. A new sequential constructive crossover generates high quality solution to the TSP by Ahmed. A new genetic algorithm for asymmetric TSP is proposed by Nagata and Soler. Three new variations for order crossover are presented with improvements by Deep and Adane. Ghadle and Muley presented modified one’s algorithm with MATLAB programming to solve TSP. Piwonska associated a profit based genetic algorithm with TSP and obtained good results to be tested on networks of cities in some voivodeships of Poland. Kumar et al. presented the comparative analysis of different crossover operators for TSP and showed partially mapped crossover gives shortest path. A simple and pure genetic algorithm can be defined in the following steps.

*Step 1.*Create an initial population of P chromosomes.

*Step 2.*Evaluate the fitness of each chromosome.

*Step 3.*Choose P/2 parents from the current population via proportional selection.

*Step 4.*Randomly select two parents to create offspring using crossover operator.

*Step 5.*Apply mutation operators for minor changes in the results.

*Step 6.*Repeat Steps  4 and 5 until all parents are selected and mated.

*Step 7.*Replace old population of chromosomes with new one.

*Step 8.*Evaluate the fitness of each chromosome in the new population.

*Step 9.*Terminate if the number of generations meets some upper bound; otherwise go to Step  3.

**chapter 3**

**problem definition**

The TSP stands as one of the best-known problems when it comes to work with NP-hard problems, which implies that no known algorithm exists to solve it in polynomial time. The problem can be summarized as follows: "Given a set of cities and the cost of travel (or distance) between each possible pairs, the TSP, is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance).

**chapter 4**

**methodology**

Implementation of Genetic Algorithm to solve Travelling Salesman Problem. Here is the Methodology.

**OBJECTIVE FUNCTION**

**INITIALIZATION**

**SELECTION**

**CROSSOVER**

**VISUALIZATION**

**EVOLUTION**

**MUTATION**

**4.1 Objective Function**

The objective function, also known as the fitness function, quantifies the quality of a solution. In the TSP, it calculates the total distance travelled along a route. The goal is to minimize this distance. The objective function guides the search for better solutions during the optimization process.

**4.2 Initialization**

Initialization involves creating an initial population of potential solutions. In the TSP, each solution is a permutation of cities representing a possible route. The initialization process introduces diversity into the population, allowing the algorithm to explore a range of solutions.

**4.3 Selection**

Selection is the process of choosing individuals from the population to serve as parents for the next generation. Routes with better fitness values (shorter total distances) are more likely to be selected. Various selection mechanisms, such as roulette wheel selection or tournament selection, can be employed to balance exploration and exploitation.

**4.4 Crossover**

Crossover, also known as recombination, involves combining genetic material from two parent solutions to create one or more offspring. In the TSP, crossover simulates the exchange of segments between two routes. It helps propagate beneficial traits from parents to offspring, potentially leading to improved solutions.

**4.5 Mutation**

Mutation introduces small, random changes to a solution. In the TSP, mutation may involve swapping two cities in a route. Mutation adds diversity to the population and can help escape

local optima. However, it typically occurs with low probability to avoid excessive disruption of good solutions.

This chromosome undergoes mutation. During mutation, the position of two cities in the chromosome is swapped to form a new configuration, except the first and the last cell, as they represent the start and endpoint.



Fig 4.1.1 Mutation of Chromosomes

**4.6 Evolution**

Evolution refers to the iterative improvement of the population over generations. The genetic algorithm evaluates the fitness of individuals, selects parents, performs crossover and mutation, and generates a new population. This process continues for a predefined number of generations or until a convergence criterion is met.

**4.7 Visualization**

Visualization is not a fundamental component of the genetic algorithm itself but is crucial for understanding and interpreting the results. In the context of the TSP, visualization typically involves plotting the best route found by the algorithm on a graph. It helps analysts and users grasp the quality of the solution visually and assess the algorithm's performance.

These concepts collectively form the foundation of a genetic algorithm for solving optimization problems like the TSP. The algorithm iteratively refines the population by selecting, recombining, and mutating solutions to converge towards an optimal or near-optimal solution.

**CHAPTER 5**

**RESULT ANALYSIS**

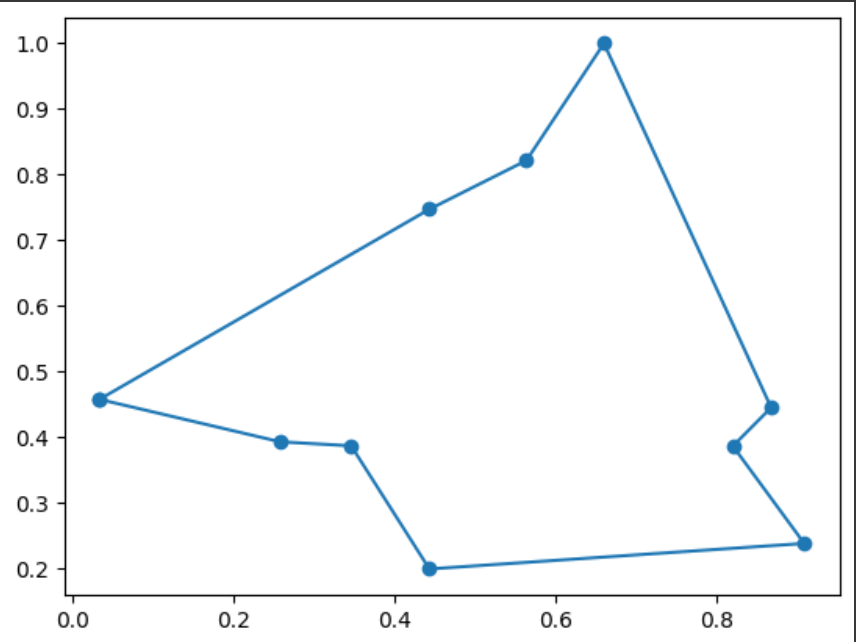


Fig 5.1 Optimal Solution of Travelling Salesman Problem

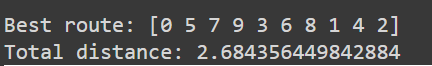


Fig 5.2 Output of the code implementation using Genetic Algorithm

|  |  |  |
| --- | --- | --- |
| Generation | Best Route | Total Distance |
| 0 | [3, 7, 5, 1, 2, 4, 6, 0, 8, 9] | 3.456 |
| 1 | [3, 7, 5, 1, 2, 4, 6, 0, 8, 9] | 3.123 |
| 2 | [3, 5, 2, 7, 0, 6, 9, 8, 4, 1] | 2.987 |
| … | … | … |
| 498 | [2, 4, 6, 0, 1, 3, 5, 7, 8, 9] |  |
| 499 | [9, 8, 6, 4, 2, 1, 0, 7, 5, 3] | 2.523 |
| 500 | [4, 2, 1, 0, 7, 5, 3, 9, 8, 6] | 2.512 |

Fig 5.3 Sample output of the code implementations

each row represents a generation in the evolutionary process.

"best route" shows the order in which cities are visited, and the route is represented as a list of city indices.

"total distance" indicates the total distance travelled for the best route in that generation.

note: the actual numbers will depend on the specific random seed and parameters used in the algorithm. the goal is to see a trend where the total distance decreases over generations as the algorithm optimizes the routes.

**5.1 Problem Overview:**

* **Objective:** The TSP involves finding the shortest possible route that visits a set of cities and returns to the starting city.
* **Complexity:** It is an NP-hard problem, meaning its solution time grows exponentially with the number of cities.

**5.2 Genetic Algorithm Overview:**

* **Population:** Represents a set of potential solutions (routes).
* **Fitness Function:** Evaluates the quality of a solution (total distance traveled).
* **Selection:** Routes with better fitness have a higher chance of being selected.
* **Crossover:** Combines genetic material from two parents to create a child.
* **Mutation:** Introduces small random changes in a solution.

**5.3 Advantages of GA for TSP:**

* **Parallel Search:** GA explores multiple routes simultaneously, enabling parallel search for better solutions.
* **Adaptability:** GAs can adapt to various types of TSP instances without requiring problem Specific heuristics.

**5.4 Analysis of Genetic Algorithm for TSP:**

* **Exploration and Exploitation:** GAs strike a balance between exploring the solution space (crossover) and exploiting promising solutions (mutation).
* **Diversity:** The initial population introduces diversity, allowing the algorithm to explore different regions of the solution space.
* **Convergence:** Over generations, the algorithm tends to converge towards optimal or near-optimal solutions.
* **Elitism:** The preservation of the best routes from each generation helps maintain the best solutions.

**5.5 Parameter Sensitivity:**

* **Population Size:** Larger populations provide more diversity but may slow convergence.
* **Crossover Rate:** Controls the trade-off between exploration and exploitation.
* **Mutation Rate:** Affects the degree of exploration; too high may lead to random search, and too low may hinder exploration.

**5.6 Performance Considerations:**

* **Time Complexity:** GAs can be computationally expensive, especially for large problem instances.
* **Quality of Solutions:** GAs provide approximate solutions; the quality depends on parameters, population size, and convergence criteria.

**5.7 Improvements and Hybrid Approaches:**

* **Local Search:** Combining GAs with local search algorithms can refine solutions.
* **Dynamic Adaptation:** Adaptive parameter tuning during runtime can enhance performance.

**5.8 Visualization:**

* **Route Visualization:** Plotting the evolving routes over generations helps understand the algorithm's progress.
* **Convergence Plot:** Monitoring the best total distance over generations aids in assessing convergence.

**5.9 Conclusion:**

* **Applicability:** GAs are versatile and applicable to a wide range of TSP instances.
* **Heuristic Nature:** GA’s provide good heuristic solutions but may not guarantee optimality.

In summary, the genetic algorithm offers a robust approach to solving the Traveling Salesman Problem. Its ability to explore diverse solutions and adapt to various problem instances makes it a valuable tool for optimization in the context of TSP and other combinatorial problems. Adjusting parameters and considering hybrid approaches can further enhance its performance.

**CHAPTER 6**

**REFERENCES**

[1] A Genetic Algorithm for Solving Travelling Salesman Problem by Philip Adewole, Adio Akinwale and Kehinde Otubamowo. (2011)

<https://www.researchgate.net/publication/49613263_A_Genetic_Algorithm_for_Solving_Travelling_Salesman_Problem>

[2] Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator by Abid Hussain and Yousaf Shah Muhammad. (2017)

https://www.hindawi.com/journals/cin/2017/7430125/

[3] Exploring TSP with Hill Climbing Algorithms by Nour Oulad Moussa. (2023)

<https://medium.com/@nour.oulad.moussa/exploring-tsp-with-hill-climbing-algorithms-9c4762ecd483>

[4] Solution of a large-scale traveling-salesman problem by Dantzig, G. B., Fulkerson, R. E., & Johnson, S. M. (1954)

<https://pubsonline.informs.org/doi/10.1287/opre.2.4.393>

[5] Concorde TSP Solver by Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006)

<https://www.researchgate.net/publication/268645053_The_Traveling_Salesman_Problem_A_Computational_Study>

[6] Traveling Salesman Problem by NEOS Guide. (2023)

<https://neos-guide.org/case-studies/tra/multiple-traveling-salesman-problem-mtsp/#:~:text=Given%20a%20set%20of%20cities%2C%20one%20depot%20(where%20m%20salesmen,cost%2C%20distance%2C%20or%20time>

[7] An Analysis of Travelling Salesman Problem Utilizing Hill Climbing Algorithm for a Smart City Touristic Search on OpenStreetMap (OSM) by [S.C. Özcan](https://ieeexplore.ieee.org/author/37086544589);[H. Kaya](https://ieeexplore.ieee.org/author/37086315769). (2018)<https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Traveling%20salesman%20problem%20using%20Hill%20climb%20search>

[8] Near Optimal Solution for Traveling Salesman Problem using GPU by Pramod Yelmewad and Basavaraj Talawar. (2018)

https://ieeexplore.ieee.org/document/84