भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay

IEOR Lab

IE684

Course Assignment Report

Instructor :

Prof. Manjesh K Hanawal

By:

Saurabh Mehra - 23M1525

Yogesh R Jangid - 23M1530

Shubham Sharad Joshi - 23M1531

Contents

1. Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) is a classic optimization problem in operations research and computer science. The problem involves a salesman who must visit a set of cities, each exactly once, and then return to the starting city. The objective is to determine the shortest possible route that allows the salesman to accomplish this.

Here's a more detailed breakdown:

- **Cities**: A set of locations the salesman needs to visit.
- **Routes**: The possible paths between cities.
- **Distance or Cost**: Each route between two cities has an associated distance or cost (e.g., time, fuel, money).
- **Objective**: Minimize the total distance or cost while visiting each city exactly once and returning to the starting city.

The TSP is an NP-hard problem, meaning that as the number of cities increases, the complexity of solving the problem grows exponentially. Exact algorithms, such as the branch and bound method or dynamic programming, can solve TSP for smaller numbers of cities. For larger instances, heuristic and approximation algorithms, like Simulated Annealing, Genetic Algorithms, or Ant Colony Optimization, are often used to find near-optimal solutions within a reasonable time.

2. Approach Used: Solving TSP with Simulated Annealing

In this section, I detail the approach I employed to solve the Traveling Salesman Problem (TSP) using the Simulated Annealing algorithm. The problem was tackled for two different instances—one with 11 cities and another with 48 cities—demonstrating the scalability and effectiveness of the method.

Problem Instances

- **11 Cities TSP**: This smaller instance allowed for initial testing and fine-tuning of the algorithm. I generated a graph representing the cities and the corresponding routes between them. The results are visually represented in an undirected graph, which illustrates the optimal path found by the algorithm.
- **48 Cities TSP**: To showcase the robustness of the Simulated Annealing approach, I scaled up the problem to 48 cities. The larger problem instance demonstrates the algorithm's ability to find near-optimal solutions within a reasonable computational time frame.

3. Simulated Annealing Algorithm

Simulated Annealing is a probabilistic technique for approximating the global optimum of a given function. It is particularly well-suited for solving the TSP due to its ability to escape local optima, which is a common challenge in combinatorial optimization problems like TSP.

The key steps in the Simulated Annealing algorithm as applied to the TSP are:

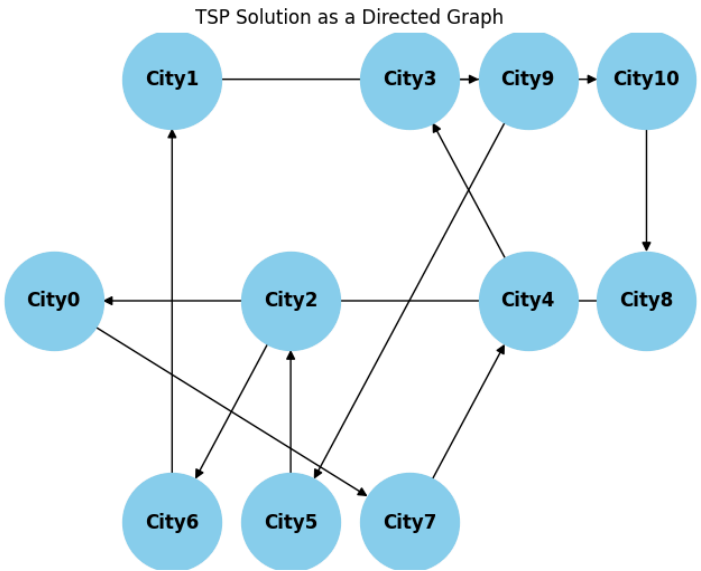
1. **Initialization**: Start with an initial tour of the cities, randomly or based on a heuristic.
2. **Temperature Schedule**: Set an initial temperature and define a cooling schedule. The temperature controls the probability of accepting worse solutions as the algorithm progresses, allowing it to explore a broader solution space.
3. **Neighborhood Search**: At each iteration, generate a neighboring solution by slightly altering the current tour (e.g., by swapping two cities).
4. **Acceptance Criterion**: Decide whether to accept the neighboring solution based on the change in tour length and the current temperature. If the new solution is better, it is always accepted; if it is worse, it is accepted with a probability that decreases as the temperature lowers.
5. **Cooling**: Gradually reduce the temperature according to the cooling schedule.
6. **Termination**: The process continues until the temperature is sufficiently low, or a predefined number of iterations is reached.

4. Results and Visualization

The results for the 11 cities problem are visualized using an undirected graph, clearly showing the optimal path determined by the algorithm. This graph illustrates the efficiency of the Simulated Annealing approach in finding a short and efficient tour that meets the TSP requirements.

For the 48 cities instance, while the complexity increased, the algorithm successfully found a high-quality solution, demonstrating its applicability to larger datasets.

For 11 cities the undirected graph is



5. MTZ Formulation for Exact Solution

The Miller-Tucker-Zemlin (MTZ) formulation is a well-known approach for solving the Traveling Salesman Problem (TSP) using Mixed Integer Linear Programming (MILP). This formulation is based on eliminating subtours (loops that do not visit all cities) by introducing additional constraints. Below, I outline the MTZ formulation used for finding an exact solution to the TSP.

Problem Definition

Given a set of cities, the objective of the TSP is to find the shortest possible route that visits each city exactly once and returns to the starting city. The MTZ formulation achieves this by using binary decision variables to model the inclusion of edges in the tour and continuous variables to eliminate subtours.

Decision Variables

- Binary Variables xij: These variables are defined for each pair of cities iii and jjj. xij = 1xij=1 if the tour includes a direct route from city iii to city jjj, and xij=0x_{ij} = 0xij=0 otherwise.
- Continuous Variables uiu_iui: These are auxiliary variables associated with each city iii, used to prevent subtours. The values of uiu_iui represent the position of city iii in the tour.

Objective Function $\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}$

The objective is to minimize the total travel cost (or distance) of the tour:

where cij is the cost of traveling from city i to city j.

Constraints

1. Tour Constraints: Ensure that each city is entered and exited exactly once:

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \quad \forall i = 1, \ldots, n$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \quad \forall j = 1, \ldots, n$$

2. Subtour Elimination Constraints: These constraints use the auxiliary variables u_i to prevent subtours:

$$u_i - u_j + n \cdot x_{ij} \leq n - 1 \quad \forall i, j = 2, \ldots, n, \; i \neq j$$

$$2 \leq u_i \leq n \quad \forall i = 2, \ldots, n$$

Here, n is the number of cities. The variables u_i represent the order in which the cities are visited, helping to ensure that the tour forms a single connected path without subtours.

6. Code and Reproducibility

The complete code and data for solving the TSP with Simulated Annealing & by MTZ formulation for both the 11 and 48 cities instances can be found in my GitHub repository. The repository includes all necessary files, detailed documentation, and instructions for replicating the results.

Github repository: https://github.com/Saurabhm4ra/Simulated-Annealing-for-TSP.git

This approach highlights the effectiveness of Simulated Annealing in solving the TSP, providing a balance between solution quality and computational efficiency, even as the problem scale increases.

7. Conclusion

The Miller-Tucker-Zemlin (MTZ) formulation is effective in solving the Traveling Salesman Problem (TSP) exactly for smaller instances, such as the 11 cities problem. The formulation guarantees an optimal solution by eliminating subtours through its set of linear constraints. However, the computational demands of the MTZ approach increase significantly with the number of cities. While it successfully provided an exact solution for the 11 cities instance, the same approach became computationally impractical for the 48 cities problem, requiring extensive time to reach a solution. This limitation highlights the trade-off between accuracy and computational efficiency in TSP formulations, particularly as the problem scale grows.