Traveling Salesman Problem using Particle Swarm Optimization

(CSN-506)

Project report submitted in partial fulfillment of the requirement of thedegree of

MASTER OF TECHNOLOGYIN COMPUTER SCIENCE AND ENGINEERING

By

Ganesh Prasad (23535005)

Pankaj Kumar Sharma (23535016)

Rajat Gaur (23535022)

Pranjal Sangal (23535033)

Under the guidance

of

Prof. Neetesh Kumar

(Professor, CSE Department)

Department of Computer Science and Engineering,Indian Institute of Technology, Roorkee

1. Problem Statement :

Traveling Salesman Problem The Traveling Salesman Problem (TSP) is one of the most famous and extensively studied combinatorial optimization problems. In the TSP, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city. The problem is often defined in the context of a salesman who needs to visit a set of cities and return to the starting city, minimizing the total distance traveled. Formally, the Traveling Salesman Problem can be defined as follows: Given: A set of n cities {1,2,...,n} The distance between each pair of cities (i,j) often represented by a distance matrix or a distance function d(i,j) Find: The shortest possible tour that visits each city exactly once and returns to the starting city. Dataset link: https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html Use any two datasets for your experiments.

2. Optimization Algorithm :

Particle Swarm Optimization

3. Implementation Details :

Traveling Salesman Problem (TSP)
The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem where the objective is to find the shortest possible route that visits each city exactly once and returns to the starting city. In our implementation, we represent the TSP as follows:

- Given a set of nn cities, numbered from 1 to nn.
- The distance between each pair of cities is represented by a distance matrix.

Particle Swarm Optimization (PSO)
PSO is a population-based stochastic optimization algorithm inspired by the social behaviour of bird flocking or fish schooling. It optimizes a problem by iteratively improving a population of candidate solutions (particles) according to a fitness function.

Implementation Overview

- **Language**: C++.
- **Libraries Used**: OpenMP for parallelization.
- **Data Structures Used**: Vectors for storing city distances and particle information.

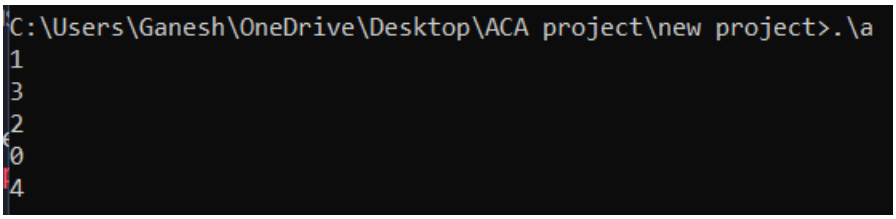
Parallel PSO Algorithm
The PSO algorithm is parallelized using OpenMP, allowing multiple particles to be evaluated simultaneously. Here's the high-level overview of the parallel PSO algorithm:

1. **Particle Representation**: Each particle represents a potential solution to the TSP. It consists of a permutation of cities and the total distance travelled in the tour.
2. **Initialization**: Initialize a population of particles randomly, each with a random permutation of cities.
3. **Fitness Evaluation**: Evaluate the fitness of each particle by calculating the total distance travelled in the tour using the given distance matrix.
4. **Global Best Update**: Update the global best tour if a particle finds a better solution.
5. **Parallel Execution**: Execute the PSO algorithm in parallel using OpenMP. Each particle evaluates its fitness and updates the global best tour within a parallel loop.
6. **Termination**: Repeat the iterations for a specified number of times or until convergence criteria are met.

4. Result

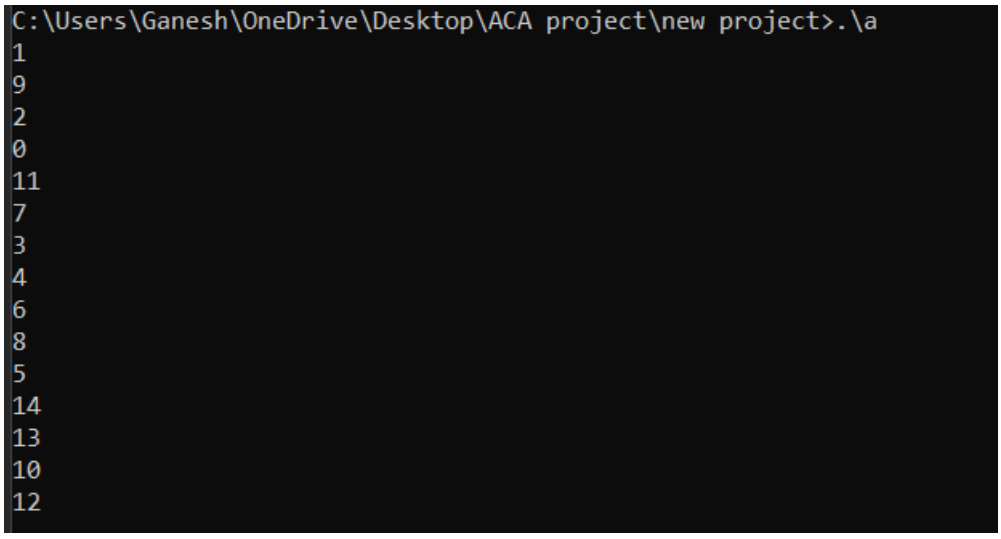
For the data set -

{0.0, 3.0, 4.0, 2.0, 7.0},
{3.0, 0.0, 4.0, 6.0, 3.0},
{4.0, 4.0, 0.0, 5.0, 8.0},
{2.0, 6.0, 5.0, 0.0, 6.0},
{7.0, 3.0, 8.0, 6.0, 0.0}



For the data set –

{0.0, 29.0, 82.0, 46.0, 68.0, 52.0, 72.0, 42.0, 51.0, 55.0, 29.0, 74.0, 23.0, 72.0, 46.0},
{29.0, 0.0, 55.0, 46.0, 42.0, 43.0, 43.0, 23.0, 23.0, 31.0, 41.0, 51.0, 11.0, 52.0, 21.0},
{82.0, 55.0, 0.0, 68.0, 46.0, 55.0, 23.0, 43.0, 41.0, 29.0, 79.0, 21.0, 64.0, 31.0, 51.0},
{46.0, 46.0, 68.0, 0.0, 82.0, 15.0, 72.0, 31.0, 62.0, 42.0, 21.0, 51.0, 51.0, 43.0, 64.0},
{68.0, 42.0, 46.0, 82.0, 0.0, 74.0, 23.0, 52.0, 21.0, 46.0, 82.0, 58.0, 46.0, 65.0, 23.0},
{52.0, 43.0, 55.0, 15.0, 74.0, 0.0, 61.0, 23.0, 55.0, 31.0, 33.0, 37.0, 51.0, 29.0, 59.0},
{72.0, 43.0, 23.0, 72.0, 23.0, 61.0, 0.0, 42.0, 23.0, 31.0, 77.0, 37.0, 51.0, 46.0, 33.0},
{42.0, 23.0, 43.0, 31.0, 52.0, 23.0, 42.0, 0.0, 33.0, 15.0, 37.0, 33.0, 33.0, 31.0, 37.0},
{51.0, 23.0, 41.0, 62.0, 21.0, 55.0, 23.0, 33.0, 0.0, 29.0, 62.0, 46.0, 29.0, 51.0, 11.0},
{55.0, 31.0, 29.0, 42.0, 46.0, 31.0, 31.0, 15.0, 29.0, 0.0, 51.0, 21.0, 41.0, 23.0, 37.0},
{29.0, 41.0, 79.0, 21.0, 82.0, 33.0, 77.0, 37.0, 62.0, 51.0, 0.0, 65.0, 42.0, 59.0, 61.0},
{74.0, 51.0, 21.0, 51.0, 58.0, 37.0, 37.0, 33.0, 46.0, 21.0, 65.0, 0.0, 61.0, 11.0, 55.0},
{23.0, 11.0, 64.0, 51.0, 46.0, 51.0, 51.0, 33.0, 29.0, 41.0, 42.0, 61.0, 0.0, 62.0, 23.0},
{72.0, 52.0, 31.0, 43.0, 65.0, 29.0, 46.0, 31.0, 51.0, 23.0, 59.0, 11.0, 62.0, 0.0, 59.0},
{46.0, 21.0, 51.0, 64.0, 23.0, 59.0, 33.0, 37.0, 11.0, 37.0, 61.0, 55.0, 23.0, 59.0, 0.0}



5. Conclusion

In this project, we implemented a Parallel Particle Swarm Optimization (PSO) algorithm to solve the Traveling Salesman Problem (TSP). The TSP is a well-known combinatorial optimization problem where the goal is to find the shortest possible route that visits each city exactly once and returns to the starting city.

Our implementation utilized the PSO algorithm, which is a population-based stochastic optimization technique inspired by the social behaviour of birds flocking or fish schooling. We parallelized the PSO algorithm using OpenMP, enabling multiple particles to be evaluated simultaneously, thus improving the efficiency of the solution search.

By employing PSO, we were able to effectively explore the solution space and find near-optimal solutions to the TSP. The parallelization allowed us to exploit multicore architectures, speeding up the optimization process and enabling the handling of larger problem instances.

In conclusion, our implementation demonstrates the effectiveness of parallel PSO in solving complex optimization problems like the TSP. The project provides valuable insights into the practical application of parallel optimization algorithms and their potential for addressing real-world challenges in logistics, transportation, and beyond.