MANIPAL UNIVERSITY
INSPIRED BY LIFE

School of Computer Science and Engineering

Department of Computer Science and Engineering

Solving the Travelling Salesman Problem using Agentic AI

Submitted By:
ADITYA SHARMA
2427030446

Supervised By:
DR. SUNITA SINGHAL

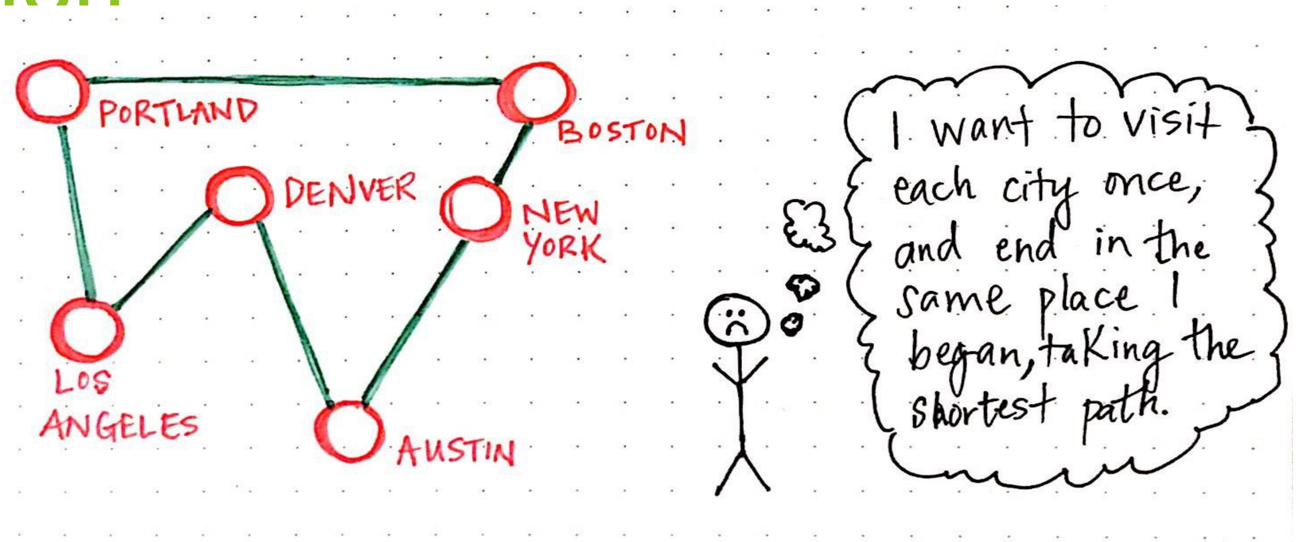
Outline:

- Introduction
- Literature Review
- Problem Statement
- Proposed Solution
- Objectives
- Results

Introduction

- The Travelling Salesman Problem (TSP) is a classical combinatorial optimization problem that aims to find the shortest possible route that visits each city exactly once and returns to the starting city.

Introduction



- TSP is classified as an NP-hard problem because the number of possible routes grows factorially with the number of cities (n!). As the problem size increases, traditional exact algorithms become computationally infeasible.

Literature Review

- TRADITIONAL METHODS:
  - Exact algorithms: Brute Force, Dynamic Programming, Branch & Bound
  - Guarantee optimal solutions
  - Computationally infeasible for large-scale TSP

Literature Review

▶ SOFT COMPUTING:

- Metaheuristics:
  - Genetic Algorithm (GA)
  - Ant Colony Optimization (ACO)
  - Particle Swarm Optimization (PSO)
- Provide near-optimal solutions efficiently
- Suitable for large problem sizes

Literature Review

AGENTIC AI:

- Agentic AI coordinates autonomous agents
- Enables task decomposition and solver orchestration
- Frameworks like **jMetal** support scalable optimization *[1]*
- Effective for hybrid TSP solving *[2]*

Problem Statement

- Finding optimal routes for large-scale TSP is computationally difficult
- Traditional methods lack scalability and adaptability

Objectives

- To apply **Agentic AI** for solving large-scale Travelling Salesman Problems.
- To enable **autonomous planning and optimization** using multiple intelligent agents.
- To use **soft computing algorithms** for efficient route finding.
- To achieve **faster and adaptive** solutions compared to traditional methods.

Proposed Solution: Overview

- Agentic AI-based hybrid optimization framework[2] [3]
- Central agent is designed to orchestrate the overall optimization process

Proposed Solution: Decomposition

- The proposed framework envisions decomposing cities into smaller clusters
- Reduces computational complexity
- Designed to enable parallel optimization

[2]

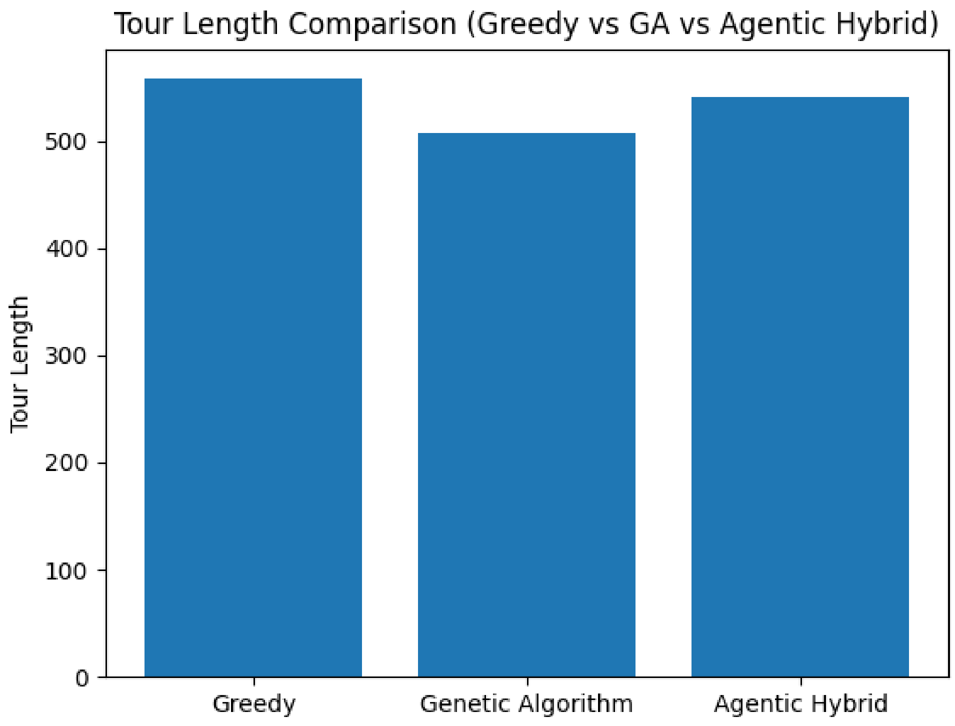
Proposed Solution : Local Optimization

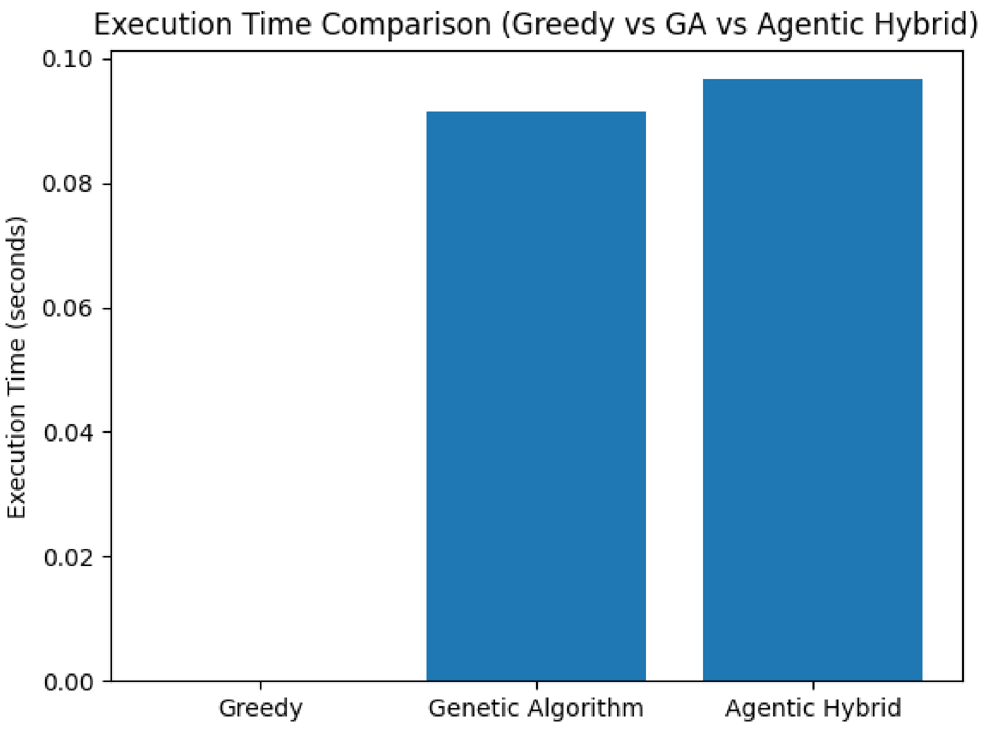
- Optimization agents can apply:
    - GA
    - ACO
    - PSO *[1]*
- Near-optimal routes are targeted for each cluster

Proposed Solution: Coordination & Learning

- Coordinator agent is designed to merge local routes
- Evaluates global tour quality
- Feedback can be used for self-improvement *[2][3]*

RESULTS:


Tour Length Comparison (Greedy vs GA vs Agentic Hybrid)
Tour Length
0
100
200
300
400
500
Greedy
Genetic Algorithm
Agentic Hybrid


Execution Time Comparison (Greedy vs GA vs Agentic Hybrid)
Execution Time (seconds)
0.00
0.02
0.04
0.06
0.08
0.10
Greedy
Genetic Algorithm
Agentic Hybrid

RESULTS:

- A Python-based simulation was developed for the Travelling Salesman Problem with **N cities**.
- Greedy and Genetic Algorithm (GA) approaches were implemented and experimentally evaluated.
- The results show that **Greedy provides the fastest execution time**, while **GA produces shorter tour lengths**.
- An **Agentic AI-based hybrid approach** was designed by combining Greedy initialization with GA refinement to balance **solution quality and execution time**.

OUTCOME:

- Designed a **agentic hybrid approach** for solving the classical TSP using Agentic AI as an orchestrator.
- Demonstrated that no single algorithm performs best in all scenarios, motivating a **dynamic hybrid solution**.
- Developed a working simulation framework in Python for experimentation and comparison.
- By next semester, the system will achieve **improved solution quality and adaptive performance** compared to standalone algorithms.

References:

[1] C. A. C. Coello, J. J. Durillo, and A. J. Nebro, “jMetal: A Java framework for multi-objective optimization,” *Advances in Engineering Software*, vol. 39, no. 10, pp. 937–951, 2008.

[2] M. Wooldridge, *An Introduction to Multi-Agent Systems*, 2nd ed. Chichester, U.K.: Wiley, 2009.

[3] OpenAI, “Agentic AI: Autonomous and tool-using systems,” Technical Overview, 2024.

Thank You