CSE331 Assignment Manual #2

Solving the Traveling Salesman Problem

1 Introduction

- This document provides the guidelines for the 2nd assignment of the 2025 CSE331 course.
- This assignment accounts for 15% of the final grade.
- The topic of the assignment is the classic computational problem named as **Traveling Salesman Problem (TSP)** [Matai et al.(2010)Matai, Singh, and Mittal].
- Students are required to implement at least **two existing algorithms**, and also design and implement **a novel algorithm of their own** from scratch.
- Students must analyse all algorithms, and compare their **runtime** and **accuracy (or approximation quality)** based on the datasets.
- Students who perform exceptionally well may receive **bonus points**, awarded based on the quality of their implementation, creativity in algorithm design, and depth of analysis.
- The use of **LLMs** (Large Language Models) is encouraged, not for direct answer generation, but as a support tool for tasks such as idea brainstorming and code refactoring. Thus, direct AI-generated solutions are not acceptable.

2 Deadline

The deadline for the assignment is **17:59 on June 10, 2025**. Please make sure to upload your report in advance, as the server may close while uploading exactly at 59 minutes. **Late submissions will not be accepted unless there are exceptional and verifiable circumstances.**.

3 Requirements

- **Survey existing algorithms and implement them**:

- Survey at least two algorithms for solving the TSP, understand their principles, and implement them from scratch *(you cannot use any library for this)* (1) MST-based 2-approximation algorithm (refer to CLRS, 3rd Edition, Section 35.2.1)[1] (2) Held-Karp algorithm [Held and Karp(1962)];
- If time permits, you are encouraged to explore and implement additional algorithms, such as Greedy heuristics, Insertion methods, Simulated Annealing, or Branch and Bound techniques.
- It is mandatory to include the algorithmic procedures and a detailed analysis.
- Students may use any programming language.

- **Design a new algorithm**:
  - It is required to design a new algorithm by extending existing techniques or applying a novel idea.
  - A clear description of the **motivation** and design philosophy is mandatory, along with an analysis and a pseudo description.

- **Experiments**:
  - Compare the algorithms based on theoretical time complexity, actual runtime, and accuracy (or approximation quality of the tour cost).
  - All algorithms must be evaluated using the same input datasets, and performance must be compared against the ground-truth results.
  - **Note on NP-hardness:** Since the TSP is an NP-hard problem, most algorithms may not scale effectively to very large instances. Students are encouraged to apply practical memory bounds or time limits during execution. If an algorithm fails to terminate within a reasonable resource limit, it is acceptable to omit the corresponding experimental results. However, please ensure that not all experiments are omitted; at least some empirical results must be reported and analysed.
  -

3.1 Report Writing Guidelines (Paper Format)

- **Language**: The report must be written in English.
- **Report structure**:
  1. Introduction
  2. Problem Statement

[1] Previously, we recommended the implementation of the Christofides heuristic [Christofides(2022)]. Given the practical challenges of implementing this algorithm—especially its reliance on the Blossom algorithm by Edmonds ($O(n^4)$ complexity)—students will be awarded bonus points for successful completion. Further bonus points will be granted for implementing the improved $O(n^3)$ Blossom algorithm [Gabow(1976)].

3. Existing Algorithms
4. Proposed Algorithm(s)
5. Experiments
6. Conclusion

- **Page Limit**: Maximum of 4 pages (excluding references; appendices are not allowed).
- **Formatting**: A template will be provided. **Font size and margins must not be changed**.
- **Code/Data Availability**: Code and datasets must be uploaded to a public repository such as GitHub. **Submitting code or data as file attachments is strictly prohibited**. If there are concerns about early exposure, you may initially keep the repository private, but it must be made public after the submission deadline. If the repository is inaccessible on the day following the deadline, it will be considered as not shared.

4 Experiments

Various datasets may be used for experimentation; however, the following four datasets must be included in your experiments. All datasets can be found at TSPLIB[2] or the TSP Test Data (TTD) [3]. These sources also provide the ground-truth values for comparison. Your experiments must report and analyse results such as accuracy relative to the ground truth and runtime performance.

Datasets That Must Be Included

- a280 dataset (a280.tsp.gz) from TSPLIB[4]
- XQL662 (xql662.tsp) from TTD[5]
- Kazakhstan data (kz9976.tsp) from TTD[6]
- Mona Lisa TSP Challenge (mona-lisa100K.tsp) from TTD[7]

5 Additional Notes

- Identical motivations are highly unlikely. If two students submit completely identical algorithms, a score of zero may be given. You are encouraged to explore topics and receive assistance in algorithm design using LLMs; however, the core ideas must be your own. The performance of the algorithm is not the primary evaluation criterion.
- **Contact**: Send an email to the TA members.

[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
[3] https://www.math.uwaterloo.ca/tsp/data/index.html
[4] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/
[5] https://www.math.uwaterloo.ca/tsp/vlsi/index.html
[6] https://www.math.uwaterloo.ca/tsp/world/countries.html
[7] https://www.math.uwaterloo.ca/tsp/data/ml/monalisa.html

A References

- How to Use Overleaf[8]
- Example GitHub Repository[9]
- Tips for Writing Academic Papers[10]

References

[Christofides(2022)] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. In *Operations Research Forum*, volume 3, page 20. Springer, 2022.

[Gabow(1976)] H. N. Gabow. An efficient implementation of edmonds' algorithm for maximum matching on graphs. *Journal of the ACM (JACM)*, 23(2):221–234, 1976.

[Held and Karp(1962)] M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

[Matai et al.(2010)Matai, Singh, and Mittal] R. Matai, S. P. Singh, and M. L. Mittal. Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, 1(1):1–25, 2010.

[8] https://www.youtube.com/playlist?list=PLHXZ9OQGMqxcWWkx2DMnQmj5os2X5ZR73

[9] https://github.com/jyyulab/MICA

[10] https://www.youtube.com/watch?v=-m3U-JrbBBg&list=PLzZ7PPT4KK5qNzQoszfF-BDIj5LOJatu1