Travelling Salesman Problem Solver Project Report

1. Overview

This project implements an interactive tool for visualising and solving the **Travelling Salesman Problem (TSP)**. The application allows users to upload a dataset of coordinates, compute an efficient route, and watch the solution animate on a canvas. The system is built with **Vanilla JavaScript and a Web Worker** for non-blocking computation. A **Flask backend** serves the frontend files, and the application is containerised using **Docker** for easy deployment and reproducibility.

2. Application Interface

The interface consists of three key components:

- **File upload section**: Users select a CSV file containing coordinates.
- **Control buttons**: Start/stop algorithm execution, skip animation, and export the computed path.
- **Visualisation canvas**: Displays all points and animates the travelling salesman path step-by-step. A live processing indicator and total-distance display enhance the interaction experience.

3. Algorithm Used

The project uses an **enhanced Nearest Neighbour heuristic** combined with a **2-Opt optimisation step**:

1. **Nearest Neighbour**
    - Iterates through all possible starting cities
    - Builds a greedy path by repeatedly visiting the closest unvisited point
2. **2-Opt Improvement**
    - Iteratively improves the initial path by reversing segments
    - Reduces unnecessary crossings and shortens total distance

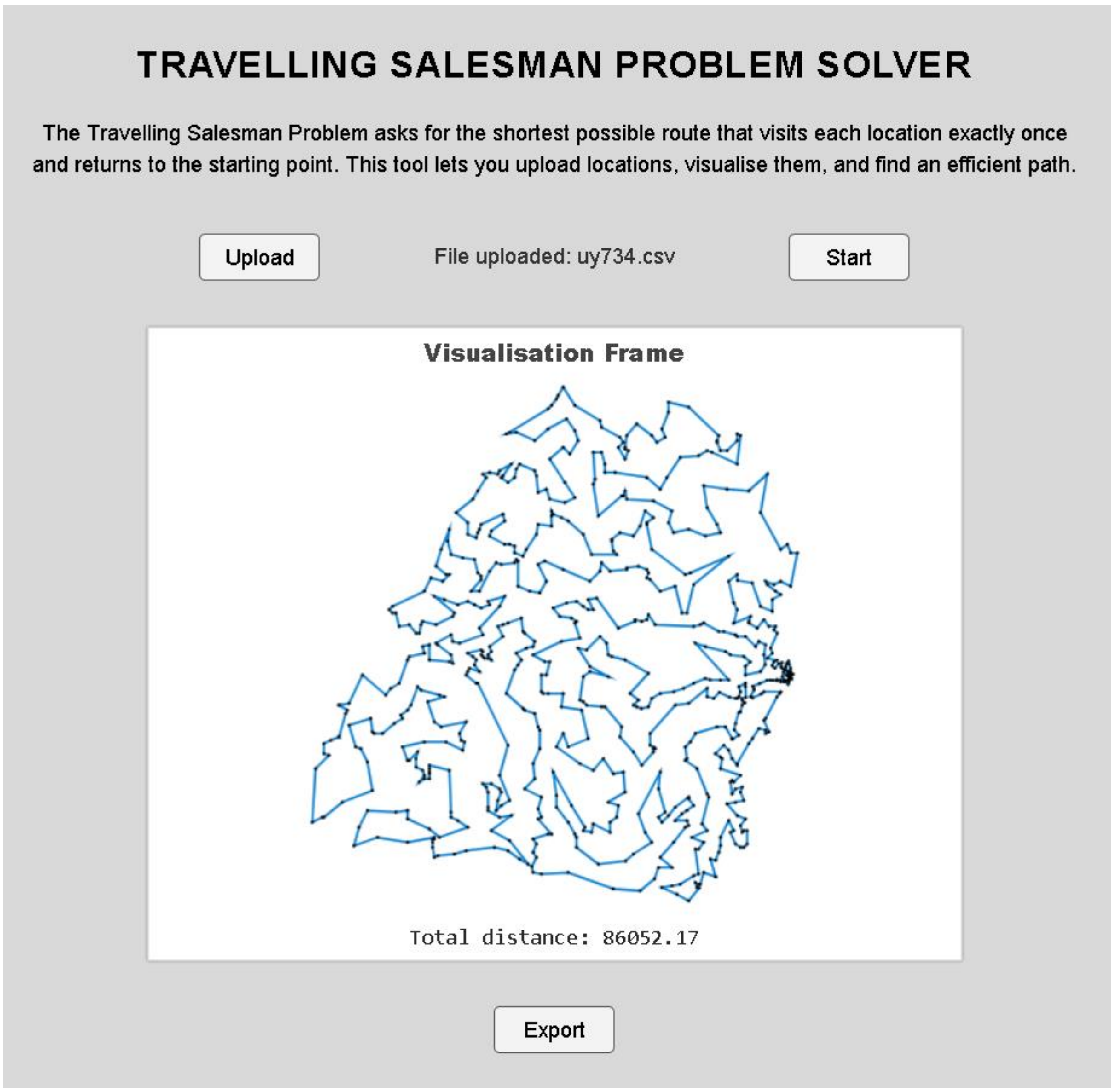
This approach provides **fast, reasonably efficient** solutions suitable for medium-sized TSP instances while keeping computation smooth in the browser via Web Workers.

4. Dataset

For demonstration and evaluation, the project uses the **Uruguay TSP dataset** (734 cities) from:

https://www.math.uwaterloo.ca/tsp/

The file contains coordinates for all 734 cities in Uruguay, making it a good medium-sized dataset for testing heuristic methods and visualising large tours.



The Travelling Salesman Problem asks for the shortest possible route that visits each location exactly once and returns to the starting point. This tool lets you upload locations, visualise them, and find an efficient path.

File uploaded: uy734.csv

Total distance: 86052.17

5. Conclusion

This project provides a simple and intuitive way to explore the Travelling Salesman Problem by combining browser-based visualisation, Web Worker computation, and a lightweight Flask backend. With full Docker containerisation and an easy interface for uploading data, running the solver, and exporting results, the application is fast to run, simple to deploy, and effective for demonstrating heuristic TSP solutions.