

# CSI 5137A – AI-enabled Software Verification and Testing

## Assignment 1, Autumn 2022

Due date: Oct 14, 2022

The assignment files have to be submitted on BrightSpace by Oct 14th midnight. The assignment can be done in groups of two or individually.

### 1 Aim

Implement a solver for the Travelling Salesman Problem (TSP) ([https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem)) using one of the metaheuristic search algorithms (e.g., Hill Climbing Search, Simulated Annealing Search, Tabu Search, Genetic Algorithm) that are discussed in the lectures or an algorithm that you research yourself and are interested to use. If you are unsure about your choice of algorithm, you can check your algorithm with me before implementing it.

### 2 Travelling Salesman Problem

The goal is to solve TSP instances as well as possible. Problem instances in TSPLIB can be found here: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>. Several datasets for TSP are available at <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>. The structure of the datasets are explained at: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf>. Among datasets that are available at <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>, it is sufficient that you focus on data format of Symmetric traveling salesman problem (TSP) (TYPE : TSP) and the edges weights are Euclidean distances in 2D (EDGE\_WEIGHT\_TYPE: EUC\_2D). More specifically, you can focus on the subset that has the following two lines in its header part (e.g., a280.tsp):

```
EDGE_WEIGHT_TYPE : EUC_2D
```

```
TYPE : TSP
```

Your program should take the .tsp file (with TYPE : TSP and EDGE\_WEIGHT\_TYPE: EUC\_2D) exactly as it is downloaded from the above site as input, and create a single output file, named `solution.csv`. It should contain a single column of city (node) indices, in the order of your solution to the TSP. Also, you should print out the total distance travelled on the standard output. For example, if the solution is to visit cities in the order of 5, 4, 1, 3, and 2, and the distance travelled is 8934.12, a python example would be like:

```
> python tsp_solver.py aaa.tsp
8934.12
> cat solution.csv
5
4
1
3
2
>
```

### 3 Deliverables

Each person should submit on BrightSpace the following deliverables by the submission deadline:

- **Implementation:** The implementation of your TSP solver. Please make sure to submit all your implementation files in separate directory and further include a readme files that explains how your program can be compiled and executed. You may implement your TSP solver in Java or Python. You can also use other language as long as you provide sufficient instructions as to how your program can be compiled and executed.

- **Report:** Include a detailed and self-explanatory report that contains a description of your solution and motivate how you approached solving this problem. In particular, describe the optimisation you have implemented in as much detail as possible. There is no page limit. Your report should be submitted as a .pdf file.

Please note that your submission should be self-contained. It should not depend on any file outside the submitted directory, such as files on your own hard drive or online. We expect the solvers simply to work out of the box. In addition, make sure to do reasonable documentation, so that I can understand and judge your solution and grade your work.