Project : Traveling Salesman Problem

Milestone : Draft Design

Due : 11/20 11:59pm PST

Length : minimum – 4 pages, maximum – 10 pages (excluding title page and references page(s)).

Submission requirements : On the title page, list all team members (names, UCI student IDs). Only the first team member listed needs to submit.

Parts I and II : Note that there are 2 parts to this project – you are required to implement an BnB DFS algorithm and a SLS algorithm for the TSP.

Overview : In this assignment, you are asked to submit a software design for the AI (smarts) Traveling Salesman Problem (https://en.wikipedia.org/wiki/Travelling_salesman_problem).

You need to

1. Design the algorithm(s) and data structures
2. Present the algorithms using pseudo-code
    a. Note – you should present the main functions of your design, even if you use a known algorithm.
    b. However, when using standard data structures, e.g. heap, priority queue, a tree, etc., you do not need to present the implementation of their member functions.
3. Explain/describe the (pseudo-code of the) algorithms
4. Provide assessment/evaluation of the time/space complexity of your algorithms.

For the Draft Design you only need to cover the AI part of the system; you can leave out other parts, e.g. input/output, etc.

You should provide sufficient detail of your design so that a person who

1. Is a competent programmer/software developer,
2. Is familiar with Data Structures and Algorithms,
3. Has taken the 271 class and earned an A

can faithfully implement your design, solely based on your (written) description/explanation, without talking to you.

Minimum AI requirements : In order to successfully complete this project, you have to apply methods and techniques you have learned in this class to provide a design for an algorithm to solve the Traveling Salesman Problem that is reasonably intelligent. For example, your algorithm is NOT reasonably intelligent if it satisfies the following condition

It uses a trivial (0) heuristic

or

It uses a SLS that effectively makes blind random moves in the state space

If your design fails the minimum AI requirement, your score will be reduced by 50%.

Criteria for grading :

1. Is the pseudo-code correct/complete (i.e. will it work)? – 50%
2. Is the description/explanation of the pseudo-code satisfactory? – 50%

Notes :

- Major AI components of your system you refer/discuss should have a formal pseudo-code definition (you should use the same level of detail as pseudo-code in the textbook, as well as an informal description/explanation in text.
- You are well advised to follow a modular design of your software. Try to break it into meaningful self-contained units that perform a particular function and that can be utilized as subroutines from other modules.
- Make sure to declare/describe any data (structures) you use. You can assume that the reader is familiar with basic data structures, e.g. priority queue, hash table, etc. and you can skip the description of these.
- The key to the BnB DFS algorithm is the heuristic function? If might be a good idea to define it as a separate (pseudo-code) module.
- In the Stochastic Local Search algorithm, does it contain a greedy element, a random element? What is the objective function guiding it? How do you decide when to stop it?