Knowledge Representation - W3 Lab: Uninformed and Informed Search

Ruairi D. O'Reilly

I. Introduction

This weeks lab is based on the lecture on Problem Solving through Search and Beyond Classical Search [1]. This weeks lab aims to enhance your understanding of uninformed and informed search strategies from both a practical and theoretical perspective.

A. Prep-work

Review and complete the following:

- search.ipynb - Jupyter notebook
- search.py - underlying library containing majority of implementation code.

Note: Your solution should be generated in Spyder as a .py file importing code as required from the AIMA repo (See the file provided for an example of importing it form a sub-directory). I will not accept .ipynb submissions and your code should import what it needs from a parent directory (no libraries to be submitted along with solutions).

B. Missionaries and Cannibals - Uninformed search

The missionaries and cannibals problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968).

- Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.
- Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?
- Why do you think people have a hard time solving this puzzle, given that the state space is so simple?

You can include your answers to the questions as comments in the code - use a simple text based representation for the diagram of the state space.

C. 8-puzzle & 8-queens - informed search

Generate a large number of 8-puzzle and 8-queens instances and solve them (where possible) by hill climbing (steepest-ascent and first-choice variants), hill climbing with random restart, and simulated annealing. Measure the search cost and percentage of solved problems and graph these against the optimal solution cost. Comment on your results.

II. Submission

Submit your solution by the due date as a single ".py" file using the following naming convention.

"W<Week_num>_Lab_<Surname>_<First name>_<Student Number>.py"

e.g.

"W3_Lab_OReilly_Ruairi_R123456.py"

References

[1] S. Russell and P. Norvig, "Ai a modern approach," Learning, 2005.