Artificial and Machine Intelligence 251/551

Semester 1, 2010

Assignment

*The written part of the assignment will be handed in by **12:00pm** on Monday **the 15th of May, 2013** - use the unit's pigeon hole on the 3rd floor in the Computer Science building. The programming part of the assignment will be electronically collected on **May the 15th, 2013** at **9:00am**.*
*The demos of the assignment will be held during the tutorials starting from the week beginning on the **20th of May**.*

PART 1 (70% of the assignment mark)

Informed search strategy implementation

Objective

The first objective of the assignment is to get practical experience in implementing two informed search strategies.

Your Task

You are required to implement the greedy search and the branch and bound search strategy for finding the path between two user specified nodes in an undirected graph. Please note that the branch and bound search will be worth 50% of the assignment mark while the greedy search will be worth 20% of the assignment mark.

The implementation should handle general graphs. The graph information will be available in the same format as in the tutorials (tutorial 2).

For the beam version of greedy search, the program should prompt the user for the start and end node in the graph. The program should return the solution path between the two nodes.

For the branch and bound search with dynamic programming, the program should prompt for the start and end node in the graph and it should return the solution path between the two nodes as well as a list of the alternative solutions and partial paths.

PART 2 (30% of the assignment mark)

Local Search Strategy Implementation

Objective

The second objective of the assignment is to get practical experience in implementing a local search strategy.

Your Task

You are required to implement an agent that performs the simulated annealing hill climbing search.

The implementation should handle all 3D maps where the map information is provided in the form of a sequence of x, y, z values. The size of the map will not be known – you can only assume that the map information is complete (there will always be x, y and z values provided for each point on the map). The program should prompt the user for the name of the file containing the map and return the following:

1) the highest elevation point on the map (the point with the highest z value),
2) the sequence of points visited to reach the highest point on the map – the sequence of points should be saved in a file called seq.txt, and the format should be exactly as the map file (text format containing the x, y, z coordinates visited in the exact order that the agent used to explore the map).

You will be provided with a sample map to test your program – the map will be available from the unit's web page. However, you are expected to test your program with other 3D maps in order to be able to accurately evaluate its performance on general 3D maps. Finally, if you wish to display the map you can use the gnuplot program available on the lab machines. You will need to use the splot function to plot the points contained in the map file as shown in the example below (assuming that you are already running gnuplot and the file map1.txt is in the current directory):

>splot "map1.txt" with points

You are required to submit the following:

ON THE DUE DATE, THE 15th of MAY

1) A two-three page report (10 point font) that specifies the following:

- any problems, bugs that the search strategies still have and how you have tried to address these problems & bugs (at most 1/2 page)

*- evaluation of the local search performance – the evaluation should show results obtained from different maps (you can easily modify the test map provided with a text editor) and should be **objective** and outline clearly how well the agent performs when given simple (one peak) and complex maps (multiple peaks).*

2) The source code (fully documented), binaries and Makefile for your system implementation. The code should be placed in your home directory under the subdirectory ami300assignment (the code should be placed in ~/ami300assignment where ~ indicates your home area). The code should run on the machines Ark/Moses and it should run without any modifications. Failure of the code to run will result in a mark of 0 for the demonstration. The name of informed search strategy binaries should be ***greedy-search*** *and* ***branch-search*** *while the name of the local search strategy should be* ***simanneal-hill****.*

The programs should not depend on any files in your home area *– it will be tested in a specially setup directory.*

NOTE!

1) The programs should be ready by the 8th of May – the last week should be spent on testing the programs.

2) Failure to put the files in the correct directory will result in a 0 for your demo part of the assignment (which represents 90% of the assignment mark).

3) You are also required to attend the last three tutorials to demonstrate your work. Failure to demonstrate your work will result in a mark of 0 for the demo part of the assignment (which represents 90% of the assignment mark).

4) YOU MUST MAKE A REASONABLE ATTEMPT TO COMPLETE THE ASSIGMENT – SUBMITTING A TITLE PAGE WITH A STATEMENT INDICATING THAT YOU COULD NOT GET ANY OF THE PROGRAMMING PART OF THE ASSIGNMENT DONE WILL NOT BE CONSIDERED A VALID ATTEMPT AND WILL RESULT IN A MARK OF ZERO FOR THE ASSIGNMENT.