UNSW COMP3431: Robot Architectures S2 2006

Assignment #1

Search in discrete and continuous spaces

Due: Start of Lab, Week 6 (1pm, 30 August 2006)

1 Overview

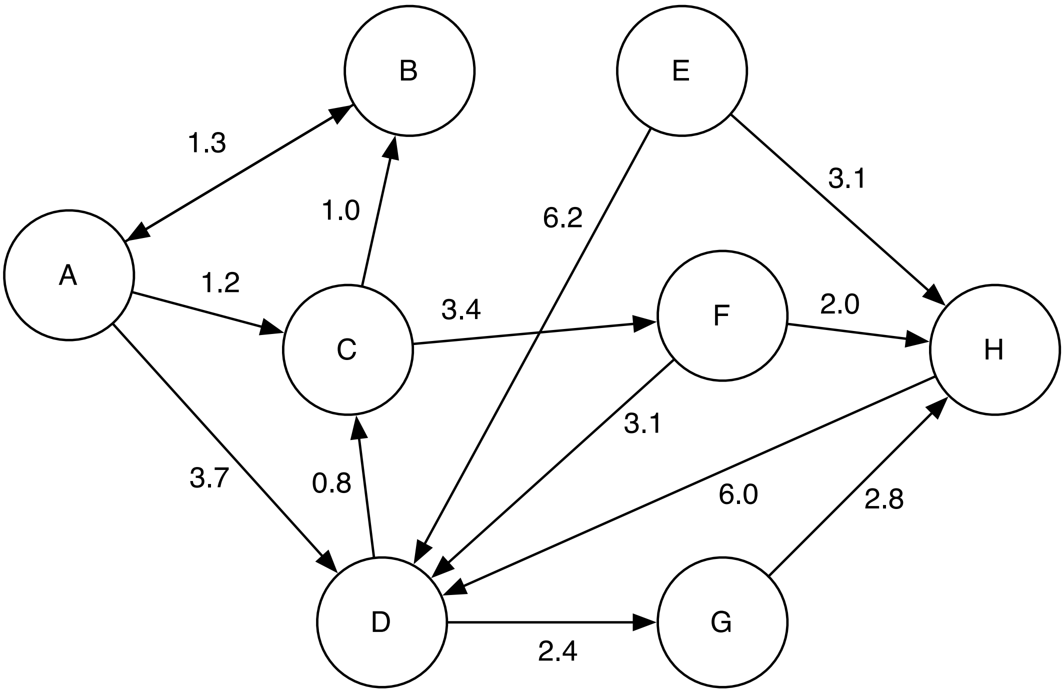
The goal of this assignment is to test your understanding of various search algorithms. While this assignment does not directly ask you to implement or trace all the algorithms discussed in class, it may be advantageous for you to do so. To answer the questions in this assignment, you will have to do some investigation of the algorithms outside of what was explicitly mentioned in class.

You should feel free to ask the Lecturer questions and discuss the assignment with others, but you should not copy someone else's work. (e.g. You can get someone to show you how to answer a question, but you should write your answer up yourself from what you remember at least 30 minutes after they've left.)

Total marks for this assignment: 90. It will be scaled to 15% of the comp3431 grade.

2 Discrete Search

In the following questions we'll be considering this graph:



With the following heuristic costs to node H:

Node	Distance
A	6.0
B	4.0
C	5.1
D	4.5
E	3.0
F	2.0
G	2.5
H	0.0

You should assume that for any node, if we need its children, we can iterate through them in alphabetical order. When asked to trace a discrete search algorithm, you should list the state of the fringe data structure just before a node is removed from it, you should also list what node is removed, and any state associated with that node. This should be repeated from when the start node is added to the fringe, until the algorithm is complete.

2.1 Question 2.1 (5 marks):

Write the pseudo-code for a generalised discrete search algorithm. Your algorithm should print out the final path between the start and goal nodes.

2.2 Question 2.2 (5 marks):

Trace the behaviour of Depth First Search on the graph starting at node A, with a goal of node H.

2.3 Question 2.3 (5 marks):

Trace the behaviour of Breadth First Search on the graph starting at node A, with a goal of node H.

2.4 Question 2.4 (5 marks):

Trace the behaviour of Best First Search on the graph starting at node A, with a goal of node H.

2.5 Question 2.5 (5 marks):

Trace the behaviour of Uniform Cost Search on the graph starting at node A, with a goal of node H.

2.6 Question 2.6 (5 marks):

Trace the behaviour of A* Search on the graph starting at node A, with a goal of node H.

2.7 Question 2.7 (10 marks):

In normal A* search, the priority of a node, n, is: $f(n) = g(n) + h(n)$. If we add a constant coefficient, α, (greater than one) to the heuristic, we get $f(n) = g(n) + \alpha h(n)$. If the optimal path cost is C^*, is there a bound on the cost of the path found by our new search algorithm? If so, what is it?

Can you prove your bound?

2.8 Question 2.8 (5 marks):

Give an admissible heuristic estimate for the minimum number of search steps from each node to H. Your heuristic should assign non-zero values to some nodes.

2.9 Question 2.9 (5 marks):

Use your heuristic to perform Best First Search on the graph. How does use of this heuristic compare with the cost-heuristic above?

3 Continuous Search

It is possible to perform a line-search analytically in some situations. Consider a function on a two dimensional space, $z = f(x, y)$. If we restrict ourselves to the line $y = mx+c$ then we can substitute that into the original function to get $z = f(x, mx+c)$. This is now a function of one variable, x. We can differentiate and set the result to 0 to get a minimum or maximum.

For example; we start with $z = x^2 + y^2 + 2xy$, and want to search along the line $y = 3x + 7$. We substitute to get $z = x^2 + (3x + 7)^2 + 2(3x + 7)x = 16x^2 + 56x + 49$. Which we can differentiate to get $z' = 32x + 56$. The differential is 0 at $x = -56/32 = -1.75$. We can back-substitute to get $y = 1.75$ for a minimum point along the line of $(-1.75, 1.75)$.

3.1 Question 3.1 (10 marks):

Show the steps involved in Powell's Direction Set method of minimisation for the function $z = 2x^2 + y^2 + 2xy$. Assume our initial direction set consists of the $(1, 0)$ and $(0, 1)$ unit vectors, and we start minimising from the point $(2, 3)$. Use the analytic method above to perform the each line minimisation.

3.2 Question 3.2 (5 marks):

For the function $z = 2x^2 + y^2 + 2xy$, what direction is conjugate to the vector $(1, 0)$?

4 Representation

Consider a small 5×5 gridworld. (i.e. a small world with 25 states laid out on a 5×5 rectangular grid.) We have four actions, each of which moves our agent to a neighbouring state, North, South, East or West. We will assume our world is deterministic. If the agent collides with a wall, then it does not move. We will assume that there are walls around the edge of the world, and between some, but not all, of the states.

There is one defined start state, and one defined goal state.

4.1 Question 4.1 (5 marks):

What is the standard transformation of this problem into a state space search?

Consider the following plan space representations of the problem:

- Partial policy: Each state in our search space is a 25-vector of discrete values. Each element of this vector is drawn from the set $\{$North, South, East, West, Undecided$\}$. The elements of the vector correspond with the actions to be taken in the states of the original problem. The vectors in this space are restricted so that vectors are only in the search space if taking the assigned actions in turn will lead us to the goal in all states for which an action is assigned.

- Full policy: Each state in our search space is a 25-vector. Each element of this vector is drawn from the set $\{\text{North}, \text{South}, \text{East}, \text{West}\}$.
- Stochastic policy: Each state in our search space is a 25×4 matrix of numbers. Each column in the matrix corresponds with a state in the original space. Each row corresponds with an action in the original space. The columns are normalised to find the probability that your agent takes that action in that state.
- Plan: Each state in the search space is a list of actions to be followed consecutively from the start state.

4.2 Question 4.2 (20 marks):

Describe a search technique that could be used with each representation. Include a description of any heuristics or evaluation functions you think are necessary.