The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2012-2013

PLANNING AND SEARCH

Time allowed TWO hours

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer FOUR out of SIX questions

Only silent, self contained calculators with a Single-Line Display are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn your examination paper over until instructed to do so

1. This question is on tree search.

   (a) Describe briefly the basic idea of tree search algorithms. Explain how a general tree search algorithm works. Your answer should explain what are the nodes and branches in a search tree and mention node expansion and search strategy. (6 marks)

   (b) Explain what it means for a search strategy to be complete. Give an example of a complete search strategy. (2 marks)

   (c) Explain what it means for a search strategy to be optimal. Give an example of an optimal search strategy. (2 marks)

   (d) Figure 1 below gives the graphic representation of cities $\{a, b, c, d, e, f, g, h, i\}$ inter-connected by highways in a country. The actual distance of the highway connecting two cities is given in km. For example, the highway for $\{a, b\}$ is 100 km in actual distance. Construct two trees generated as the result of: (i) using greedy search and (ii) A* search for the problem where the starting city is a and the ending city is i. Label the nodes expanded for both trees with the appropriate costs. For both trees, you should use the straight-line distance $v(n)$ between the city n and i: $v(a) = 300$, $v(b) = 195$, $v(c) = 120$, $v(d) = 123$, $v(e) = 125$, $v(f) = 260$, $v(g) = 235$, $v(h) = 55$, and $v(i) = 0$. Which search algorithm gives a lower total distance travelled? (15 marks)

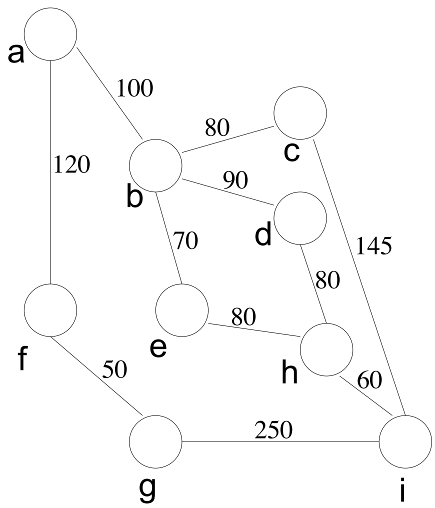


Figure 1: Cities

2. This question is on local search.

   (a) Explain how hill-climbing works. (6 marks)

   (b) Explain how genetic algorithms work. Your answer should include the terms chromosome, fitness function, crossover and mutation. (6 marks)

   (c) Consider the following search problem:

   - the set of states $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$
   - successors of s_0 are $\{s_0, s_1, s_2\}$
   - successors of s_1 are $\{s_1, s_2, s_3\}$
   - successors of s_2 are $\{s_2, s_3\}$
   - successors of s_3 are $\{s_0, s_3, s_4\}$
   - successors of s_4 are $\{s_4, s_5\}$
   - successors of s_5 are $\{s_2, s_3, s_5\}$
   - objective function f is as follows: $f(s_0) = 0$, $f(s_1) = 3$, $f(s_2) = 2$, $f(s_3) = 4$, $f(s_4) = 1$, and $f(s_5) = 5$.

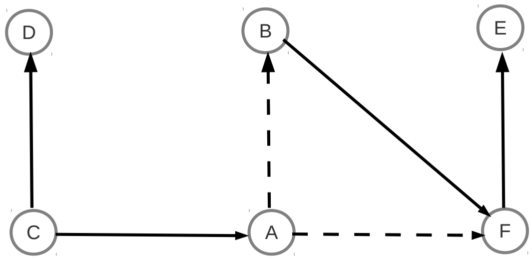
   i. Which (if any) states are local maxima and global maxima? (3 marks)

   ii. Trace hill climbing search starting in state s_0 (indicate which state is considered at each iteration, and which solution is returned when the search terminates). Does it return the optimal solution? (5 marks)

   iii. Consider a stochastic hill-climber procedure for solving the same problem. In every iteration one successor state is randomly chosen from the set of successors. If the current state's objective function is strictly higher than that of the chosen successor, the current state is returned and the procedure terminates. If the random successor's objective function is the same or higher, than it is chosen as the successor. The random successor is chosen by drawing a random number r from $[0, 1]$. If the current state has n successors, listed in the order above, each of them gets assigned an interval, $[0, 1/n], (1/n, 2/n], \ldots, (n-1/n, 1]$, and the successor state is chosen depending on which interval r belongs to. For example, for s_2 with two successors $\{s_2, s_3\}$, the intervals are $[0, 0.5]$ for s_2 and $(0.5, 0]$ for s_3, and if r is 0.3 then s_2 is chosen, and if r is 0.7 then s_3 is chosen.Trace this procedure starting in state s_0 for a sequence of r given by $(0.2, 0.4, 0.8, 0.6)$. Indicate which state is considered at each iteration, which successor state's interval r is in, and which solution is returned when the search terminates. Does it return the optimal solution? (5 marks)

3. This question is on search with non-deterministic actions.

Consider the following search problem. The set of states corresponds to a set of locations on the map (the state A corresponds to being in location A, etc.). Actions correspond to following a directed edge on the map (going from one location to another). If there is an edge from x to y then it is possible to perform the action $go(x, y)$. All actions apart from two are deterministic and have the expected result, namely going from x to y results in being in y. The only exceptions are actions which involve moving out of A: the action of moving from A to F has two possible outcomes, one being in F and another being in B. Similarly, the action of moving from A to B also has two possible outcomes, one being in B and another being in F. The initial state is being in C and the goal is to reach E.



(a) Give a list of all possible actions in state A and a list of all possible actions in state C. (2 marks)

(b) For states A and C, and every action which is possible there, define the function $Results(state, action)$. (3 marks)

(c) Draw the and-or search tree for this search problem. (7 marks)

(d) Define what is a solution to an and-or search problem. (3 marks)

(e) Draw a subtree which is a solution to the problem. (5 marks)

(f) State the corresponding conditional plan. (5 marks)

4. This question is on situation calculus.

Consider the following problem. An agent which is a robot vacuum cleaner is moving between several rooms. It can perform the following actions:

- $Suck(x)$ which means sucking up dirt in room x. This action is possible when the agent is in room x. The result of this action is that room x is clean.
- $Move(x, y)$ which means move from room x to room y. This action is possible if the agent is in room x, and y is different from x. The result of this action is that the agent is no longer in room x, and is in room y.

The fluents are: $In(x, s)$ which means that the agent is in room x in situation s, and $Clean(x, s)$ which means that room x is clean in situation s.

(a) Write possibility axioms for the actions. (6 marks)

(b) Write successor state axioms for the fluents. (10 marks)

(c) Suppose in S_0, the agent is in room A, and both rooms A and B are not clean. Find a substitution for the variables a_1, a_2, a_3 which makes the following sentence true: (5 marks)

$$Clean(A, Result(a_3, Result(a_2, Result(a_1, S_0))))\wedge$$

$$Clean(B, Result(a_3, Result(a_2, Result(a_1, S_0))))$$

(d) Explain what the frame problem is and why successor state axioms can be considered to be a solution to it. (4 marks)

5. This question is on planning in general and on goal stack planning.

(a) Define the following planning problem in Planning Domain Description language (specify predicates, objects, initial state, goal specification, action schemas). (10 marks)

> There are two new doors installed in the house: the front door and the back door. At the moment they are not painted. The agent can paint a door, provided it is in front of the door. It can also move from one door to the other. In the initial state the agent is by the front door. The goal is to have both doors painted.

(b) Trace the goal stack planning algorithm for this problem. At each step, show what the stack contains, what is the current plan, and the state of the knowledge base if it changes. (15 marks)

6. This question is on planning in general and on partial order planning.

   (a) Explain the difference between how a *classical search problem* and a *classical planning problem* are formulated, and between solutions to a search problem and a planning problem. (5 marks)

   (b) Suppose there are two objects: $BlockA$ and $BlockB$, and three predicates, $OnTable(x)$, $On(x, y)$, $Clear(x)$. State S_0 is given as follows:

   $$\{OnTable(BlockA), OnTable(BlockB), Clear(BlockA), Clear(BlockB)\}$$

      i. List all the negated ground fluents which are true in S_0. (2 marks)

      ii. Suppose there is an action $Stack(x, y)$ with the following pre- and postconditions:

      $Stack(x, y)$:
      PRECOND: $Clear(x) \wedge Clear(y) \wedge OnTable(x) \wedge OnTable(y)$
      EFFECT: $Clear(x) \wedge \neg Clear(y) \wedge On(x, y) \wedge \neg OnTable(x)$

      Give a description of the state resulting from executing $Stack(BlockA, BlockB)$ in state S_0. (3 marks)

   (c) Consider the following planning problem. The agent wants to get dressed for cold weather: put on a jacket, a coat and a hat. The predicates are: Hat (meaning, the agent has a hat on), $Jacket$ (meaning, the agent has a jacket on) and $Coat$ (meaning, the agent has a coat on). The agent cannot put on a jacket if it already has a coat on (but can put a coat on top of the jacket).

   The actions are:

   $HatOn$:
   PRECOND:
   EFFECT: Hat

   $JacketOn$:
   PRECOND: $\neg Coat$
   EFFECT: $Jacket$

   $CoatOn$:
   PRECOND:
   EFFECT: $Coat$

   The initial state is $\{\}$ (none of the predicates is true). The goal is $Hat \wedge Jacket \wedge Coat$. Solve this problem using partial order planning; trace the search from the initial empty plan to a complete solution, explaining each step. (15 marks)