CS 440: Introduction to Artificial Intelligence

Homework #2 Search Problems in AI

March 10, 2023

Professor Abdelsam Boularias

Jay Patwardhan 208001851
Alan Wu 208000574
Neel Shejwalkar 207004853

Problem 1

Tracing Operation A^* from Lugoj to Bucharest

Problem 2

Consider a state space where the start state is number 1 and each state k has two successors: numbers $2k$ and $2k+1$.

Part A

Suppose the goal state is 11. List the order in which states will be visited for breadthfirst search, depth-limited search with limit 3, and iterative deepening search.

Solution.

Part B

How well would bidirectional search work on this problem? List the order in which states will be visited. What is the branching factor in each direction of the bidirectional search?

Solution.

Problem 3

Correct vs. Not Correct

Part A
Breadth-first search is a special case of uniform-cost search.

Part B
Depth-first search is a special case of best-first tree search.

Part C
Uniform-cost search is a special case of A^* search.

Part D
Depth-first graph search is guaranteed to return an optimal solution

Part E
Breadth-first graph search is guaranteed to return an optimal solution.

Part F
Uniform-cost graph search is guaranteed to return an optimal solution.

Part G
A^* graph search is guaranteed to return an optimal solution if the heuristic is consistent.

Part H
A^* graph search is guaranteed to expand no more nodes than depth-first graph search if the heuristic is consistent.

Part I
A^* graph search is guaranteed to expand no more nodes than uniform-cost graph search if the heuristic is consistent.

Problem 4

Iterative deepening is sometimes used as an alternative to breadth first search. Give one advantage of iterative deepening over BFS, and give one disadvantage of iterative deepening as compared with BFS. Be concise and specific.

Problem 5

Prove that if a heuristic is consistent, then it must be admissible. Construct an example of an admissible heuristic that is not consistent. (Hint: You can draw a small graph of 3 nodes and write arbitrary cost and heuristic values so that the heuristic is admissible but not consistent.)

Problem 6

In a Constraint Satisfaction Problem search, explain why it is a good heuristic to choose the variable that is most constrained but the value that is least constraining.

Problem 7

Consider the following game tree, where the first move is made by the MAX player and the second move is made by the MIN player.

Part A
What is the best move for the MAX player using the minimax procedure?

Part B
Perform a left-to-right (left branch first, then right branch) alpha-beta pruning on the tree. That is, draw only the parts of the tree that are visited and don't draw branches that are cut off (no need to show the alpha or beta values).

Part C
Do the same thing as in the previous question, but with a right-to-left ordering of the actions. Discuss why different pruning occurs.

Problem 8

Which of the following are admissible, given admissible heuristics h_1, h_2? Which of the following are consistent, given consistent heuristics h_1, h_2? Justify your ans

Part A

$h(n) = min(h_1(n), h_2(n))$

Part B

$h(n) = \omega h_1(n) + (1 - \omega)h_2(n)$, where $0 \leq \omega \leq 1$.

Part C

$h(n) = max(h_1(n), h_2(n))$

Part D

Which of these heuristics, a, b, or c, would you choose?

Problem 9

Simulated annealing is an extension of hill climbing, which uses randomness to avoid getting stuck in local maxima and plateaux.

Part A

For what types of problems will hill climbing work better than simulated annealing? In other words, when is the random part of simulated annealing not necessary?

Part B

For what types of problems will randomly guessing the state work just as well as simulated annealing? In other words, when is the hill-climbing part of simulated annealing not necessary?

Part C

Reasoning from your answers to parts (a) and (b) above, for what types of problems is simulated annealing a useful technique? In other terms, what assumptions about the shape of the value function are implicit in the design of simulated annealing?

Part D

As defined in your textbook, simulated annealing returns the current state when the end of the annealing schedule is reached and if the annealing schedule is slow enough. Given that we know the value (measure of goodness) of each state we visit, is there anything smarter we could do?

Part E

Simulated annealing requires a very small amount of memory, just enough to store two states: the current state and the proposed next state. Suppose we had enough memory to hold two million states. Propose a modification to simulated annealing that makes productive use of the additional memory. In particular, suggest something that will likely perform better than just running simulated annealing a million times consecutively with random restarts. [Note: There are multiple correct answers here.]

Part F

Gradient ascent search is prone to local optima just like hill climbing. Describe how you might adapt randomness in simulated annealing to gradient ascent search avoid trap of local maximum.