Dr. Eick

Fundamentals of Artificial Intelligence

COSC 4368

Solution Sketches Midterm Exam

A

Wednesday, March 9, 2022



*Name:*

*Student id:*

1. A\* & Best-first Search & Backtracking (15 points):
2. Reinforcement Learning (12 points):
3. SA and Hill Climbing (8 points):
4. Game Theory (5 points):
5. Constraint Satisfaction Problems (11 points):
6. Miscellaneous Questions (13 points):

Σ (out of 64):

Number Grade:

The exam is “open books and notes”, but no computers and cell phones allowed; you have 72 minutes to complete the exam. Write all your answers on this document (you can use back sides!).

**1) Best-first Search and A\* [15]**

Consider the search space below, where *S* is the start node and *G1* and G2 satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal (the h function itself) is reported inside nodes.

For each of the following search strategies, indicate which goal state is reached (if any) and list, *in order*, all the states *popped off of the OPEN list*. When all else is equal, nodes should be removed from OPEN in alphabetical order.

No partial credit!

##### a) Best-First-Search (using function h only) [3]

Goal state reached: G2 [1]

States popped off OPEN: S, B, G2 [2]

##### b) A\* (using f=g+h)[4]

Goal state reached: G2 [1]

States popped off OPEN: S, B, G2 [3]

No partial credit!

5

1

7

2

1

5

2

9

2

5

8

4

1

4

7

Problem 1 continued

b) A\* terminates when the goal state is expanded but not when a goal state is pushed on the open list; explain why this is the case! Why doesn’t A\* terminate as soon as a goal state is found? [3]

Let us assume a goal state g’ is in the open list; as long as states s with lower values for g(s)+h(s)<g(g’)+h(g’) exist on the open list there is a chance that going through s are better solutions in comparison to the current path to g’. Therefore, terminating immediately when g’ is added to the open list will not always find the optimal solution for admissible heuristics h!

c) Compare Backtracking (assuming it is used with an intelligent operator selection function) with Best First search (assuming it is used with an intelligent state evaluation function). What are the advantages/disadvantages of each search strategy? [6]

Let n be the number of nodes of the search tree

d’ the maximum depth encountered in the search

|  |  |
| --- | --- |
| BT | BFS |
| Works on a single path [0.5] | Can work on multiple paths in parallel, which is advantageous for some search problems [2] |
| Only needs to store the current path and therefore has much lower storage requirements O(d) than BF2 [2] | Needs to store a large proportion of the search tree O(n) and might run out of storage for complex search problems [2] |
|  |  |
|  |  |
|  |  |

**Other answers might deserve credit!**

**2) Reinforcement Learning**

a) Consider the following World called DEF is given:



Give the Bellman equations for states 1, 2, 4 of the DEF World; assume γ=0.8! [4]

U(1) = -4 + 0.8\*max(U(2),U(4))

U(2) = 0 + 0.8\*U(3)

U(4) = -3 + 0.8\*max(U(1)\*0.1+U(5)\*0.9, U(6))

Grading guidelines: One points for each equation; one additional point if all 3 solutions are correct.



DEF World

b) Now we apply temporal difference learning, assuming the agent starts in state 1 and applies the operator sequence **n-e-sw**; assume the initial utilities are 0; what are the new utilities of the states visited by the agent; also assume α=0.5 and γ=1? [6]

UΠ (s) 🡨 UΠ  (s) + α [ R(s) + γ UΠ (s’) - UΠ (s) ]

U(1) = U(1) + 0.5[R(1) + 1\*U(2) – U(1)]

= 0 + 0.5[-4 + 0 – 0] = -2

U(2) = U(2) + 0.5[R(2) + 1\*U(3) – U(2)]

= 0 + 0.5[0 + 0 – 0] = 0

U(3) = U(3) + 0.5[R(3) + 1\*U(1) – U(3)]

= 0 + 0.5[5 + -2 – 0] = 1.5

If 1 error at most 4 points; if 2 errors 0 points. Equation correct but utility wrong: at most 4 points

c) What role does the learning rate α play in temporal difference learning? How does choosing large values for α impact the temporal difference learning algorithm? [2]

State utilities will be updated more quickly or a larger step size is used when updating state utilities

**3) SA, and Hill Climbing [8]**

a) Assume you apply randomized hill climbing using three different initial positions. What can be said about the three solutions that are found in these 3 runs? [2]

The found solution will be on the hill the initial starting position resides on (or, maybe RHC finds the local maximum---or something close to it---with respect to the initial position.

b) What are the advantages of using hill climbing approaches over best first search approaches [2]

much lower storage requirements; will never run out of storage.

Problem 3 continued

b) Simulated annealing employs a temperature parameter which is decreased based on a cooling schedule when solving a search problem. What role does the temperature play in a simulated annealing algorithm? What is the motivation for this reducing the temperature? [4]

Temperature determines the likelihood that a downward move will be accepted. [2]

Reducing the temperature/the likelihood of downward moves being accepted near the end of the run allows the algorithm to find a local maximum/minimum[[1]](#footnote-1) more quickly[2]. Having high temperatures early supports exploration in the search and often allows to escape “bad” local maximum, allowing to find better solutions. [1]

At most 4 points!

**4) Game Theory [5]**

What is the Nash Equilibrium for the following parallel game, whose payoff matrix is depicted below [3]? Player 1 has actions A, B, and C whereas Player 2 had actions D, E and F. What is the main property of a Nash Equilibrium? [2]

|  |  |  |  |
| --- | --- | --- | --- |
|  | D | E | F |
| A | 5,2 | 3,5 | 1,4 |
| B | **9**,2 | 4,4 | 3,6 |
| C | 1,**7** | 5,5 | 7,1 |

Underlined pairs of player actions are in red!

As there is no box where both choices are in red, therefore there is no Nash equilibrium for the game. because there is no best outcome for both players.

[3] no partial credit.

If one player changes its action, her reward will not decrease [2].

5. **Discrete CSPs (11 points)**

Assume a constraint satisfaction problems in which variables A, B, C, D take values in {1,…,100}

* **Constraints:**
  + (C1) B<A
  + (C2) C\*D + B\*C\*C=A\*B\*B\*C
  + (C3) C\*C\*D + A\*B\*C = A\*A\*B\*B

A brute force solution to this problem could look as follows:

FOR A=1,…,A=100

FOR B=1,…,B=100

FOR C=1,…,C=100

FOR D=1,…,D=100 DO {

IF C1 and C2 THEN WriteSolution(A,B,C,D)}

Give the code of a more efficient solution to this problem which uses less loops and/or less iterations inside the loop. Briefly describe the idea of your solution! Solutions which speed up the solution to the CSP more will get more points.

We first transform equation (C2) by dividing each side by C obtaining

(C2’) D + B\*C=A\*B\*B

From this we obtain equation (C2’’) which allows us to eliminate variable D

(C’’) D=(A\*B\*B-B\*C)

and by substituting D in equation (C3) we get equation C3’:

(C3’) C\*C\*(A\*B\*B-B\*C) + A\*B\*C = A\*A\*B\*B

Additionally considering that B<A, we obtain the following much more efficient loop to solve the above CSP:

FOR A=1,…,A=100

FOR B=1,…,B=**A-1**

FOR C=1,…,C=100 DO

{ IF C3’ THEN{ D=(A\*B\*B-B\*C); WriteSolution(A,B,C,D)}}

If they only consider eliminating variable D: 7 points

If they only take advantage of B<A: 4 points

However, they might propose other approaches to simplify the loop which might deserve partial (or unlikely full) credit.

**6) Miscellaneous Questions (13 points)**

a) Are the main characteristics of crossover operators in evolutionary computing systems? What role do they play in the search for the fittest solution? [3]

Crossover is an exploitation operator [1]. It does not introduce anything new [1], but probabilistically combines partial solutions from each parent when producing the offspring [2].

At most 3 points!

b) Evolutionary computing systems employ Darwinian Evolution / Survival of the fittest in the search for good solutions. Explain! Describe an approach that could be used to simulate Darwinian Evolutions! [4]

fitter solutions have a higher probability in participating in the breeding of the next generation [2]

e.g. Roulette wheel selection allows the fitness proportional generation of a mating pool [2] other answers might deserve 2 points as well!

c) What is the goal of reinforcement learning? [3]

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment---that might potentially change--- in order to maximize the notion of cumulative reward.

or

 In general, a reinforcement learning [agent](https://www.techtarget.com/searchenterpriseai/definition/agent-intelligent-agent) is able to perceive and interpret its environment, take actions and learn through trial and error.

Other answers might deserve full (or partial) credit!!

d) For most game with complete information, such as Go and chess, it is not feasible to construct the complete search tree; how do game-playing programs cope with this challenge? [3]

The define an evaluation function which evaluate non terminal states and then search states up to a depth bound to avoid running out of storage / unacceptable time complexities.

Be liberal in giving partial credit as long as they capture parts of the answer given above.

1. Or a solution close to it… [↑](#footnote-ref-1)