15-281 Archives | Practice Final #2

INSTRUCTIONS

- Exam length: 180 minutes
- You are permitted to have three handwritten 8.5"x11" pages of notes, double-sided
- No calculators or other electronic devices allowed

Name	
Andrew ID	

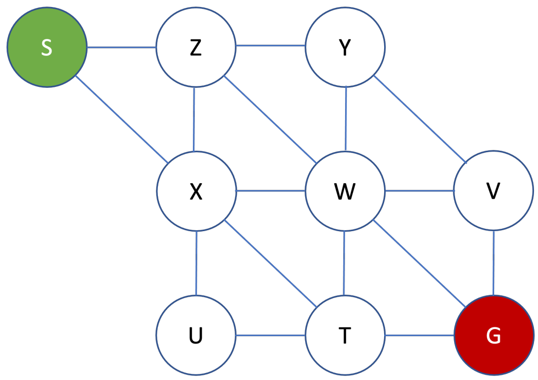
For staff use only

Q1. Search	/ 27
Q2. Linear and Integer Programming	/ 26
Q3. Local Search	/ 22
Q4. Q-Learning	/ 36
Q5. Bayes Nets and Sampling	/ 39
Q6. Hidden Markov Models	/ 26
Q7. Game Theory	/ 24
Total	/200

Q1. [27 pts] Search

Below is a graph with 9 nodes and vertical, horizontal, and diagonal edges. Suppose two robots need to traverse this graph simultaneously, starting at node S and ending at the same time at node G. A valid pair of paths ensure that they do not collide at any node (not at the same node at the same time) nor should they traverse the same edge at the same time (either in the same direction or opposite directions).

In order for you help the robots get to their goal, you need to design the number of seconds it takes the robots to traverse each edge (the edge cost). This cost should be the same for both robots and all paths to the goal. Remember: both robots must arrive at G at the same time (i.e., the sum path length must be the same for both robots). Additionally, you must design two sets of **consistent** heuristics (one per robot) that ensures that they choose **different optimal paths** if they were to run A* graph search.



(a) [9 pts] For the costs, each horizontal edge must be the same, each vertical edge must be the same, and each diagonal edge must be the same traversal time. Fill in horizontal, vertical, and diagonal edge costs that you will use for the robots. You can use the same number more than once!

horizontal	**vertical**	**diagonal**
h+v = d, we used 1	We used 1	We used 2

(b) [18 pts] Next fill in two sets of consistent heuristics for the robots such that the robots take different paths (different edges and different nodes) when using A* search. Your heuristic values can be whole numbers or fractions. For tie breaking, use alphabetical order.

Node	Robot 1	Robot 2
S	2	2
Z	4	1.5
Y	3	1
X	1.5	3
W	2	2
V	1	0.5
U	1	2
T	0.5	1
G	0	0

Q2. [26 pts] Linear and Integer Programming

A factory is making pillows and stuffed animals. They have to make at least 2 pillows and at least 2 stuffed animals. The number of stuffed animals minus the number of pillows should be no more than 3 (but there could be an unconstrained number of more pillows). A pillow takes 1 unit of stuffing and a stuffed animal takes 1/4 unit of stuffing. The factory has 10 units of stuffing. The factory wants to maximize the number of pillows and stuffed animals that they make. They cannot make partial objects.

Graph paper is provided on the next page for convenience and won't be graded. You should **NOT** need a calculator for this problem. All of the values work out pretty evenly.

(a) [6 pts] Write the constraints in matrix form below. Use P for pillows and S for stuffed animals.
Assume x = $[P, S]^T$

A:	**b:**
$[-1, 0]$ $[0, -1]$ $[1, .25]$ $[-1, 1]$	$[-2, -2, 10, 3]^T$

(b) [8 pts] What are the corners (S, P) of the constrained polygon?

Corners:
$(2, 9.5), (2, 2), (5, 2), (10.4, 7.4)$

(c) [2 pts] Formulate this problem as a minimization problem. What is the cost vector?

Cost vector:
$-P - S$

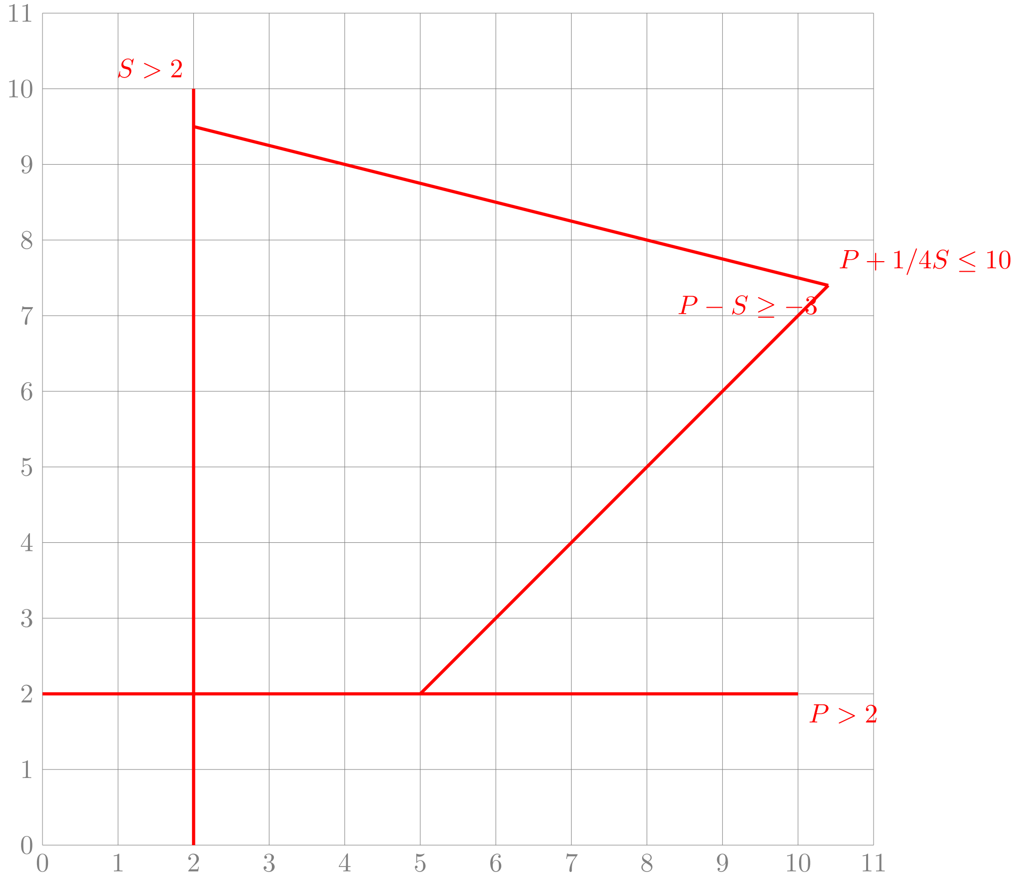
(d) [4 pts] You should find that the minimum cost corner is non-integer. Branch on S. What is the minimum objective cost of the LP problem pulled off the queue next? What is the vertex (S, P) of that minimum objective?

Minimum Cost:	**Vertex:**
Split on either side of 10.4. $S > 11$ is infeasible. -17.5 is the objective pulled off.	$(10, 7.5)$

(e) [6 pts] Run branch and bound to completion. What is the objective of the integer solution? What is the vertex (S, P) of that minimum objective?

Minimum Cost:	**Vertex:**
-17	$(10, 7)$

GRAPH PAPER FOR YOUR CONVENIENCE



Q3. [22 pts] Local Search

You are given symbols (one per word) in a language that you don't understand and a native speaker who can rate the validity of the syntax of the sentence. For example, if your English speaking friend was asked to rate sequences of the words { she, he, chess, play, and, together}, they should rate the sequence "he and she together play chess" **higher** than "he play chess she and together". Notice that some sequences may be equivalent, for example, "he and she together play chess" and "she and he together play chess" should probably both be rated equally. Words can only be used once in a sequence, and all words must be used in the sequence.

In order for you to figure out valid sequences of symbols in a new language, you decide to use local search. You start with a random sequence and improve on it.

(a) [2 pts] If n is the number of words in the sequence, what is the size of the state space for this ordering problem?

State space:
$n!$

(b) [4 pts] Define a neighbor for this problem. In other words, how will your local search algorithms determine neighboring states?

Neighbor:
Lots of valid answers. One possible answer is to swap two words in a sequence. Another could be to bring any word in the sequence to the front.

(c) [5 pts] Will traditional greedy hill climbing always find the optimal solution?

○ Yes ● No

Why or why not?

Why:
It is possible that there is a local maximum that the greedy search would stop at and not be able to find a better solution.

(d) [5 pts] Will a random walk necessarily find the optimal solution in worst-case $O(n)$ time?

○ Yes ● No

Why or why not?

Why:
There are n! different sequences of words so it may need to search all n! of them to find the best. Also the random walk may repeat states and is only guaranteed to find the optimal solution as $t \to \infty$

(e) [6 pts] Suppose you decide to try simulated annealing. Recall the algorithm below:

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
          schedule, a mapping from time to "temperature"

  current ← MAKE-NODE(problem.INITIAL-STATE)
  for t = 1 to ∞ do
      T ← schedule(t)
      if T = 0 then return current
      next ← a randomly selected successor of current
      ΔE ← next.VALUE – current.VALUE
      if ΔE > 0 then current ← next
      else current ← next only with probability e^(ΔE/T)
```

Which of the following make it most likely that simulated annealing will transition to a lower valued state, assuming we're definitely executing the last line in the algorithm? By increasing ΔE, we mean on a number line, e.g., from -2 to -1 or from 1 to 2.

- ● Increase T, Increase ΔE
- ○ Increase T, Decrease ΔE
- ○ Decrease T, Increase ΔE
- ○ Decrease T, Decrease ΔE

Should T be decreased slowly (more timesteps t) or quickly (fewer timesteps t) to increase the chances that it finds the optimal solution?

● Slowly ○ Quickly

Q4. [36 pts] Q-Learning

(a) Consider a system with two states and two actions. You perform actions and observe the rewards and transitions listed below. Each step lists the current state, action, resulting state, and reward as S, A, S', R. Perform Q-learning using a learning rate of $\alpha = 0.5$ and a discount factor of $\gamma = 0.5$ for each step. **The Q-table entries are initialized to zero.**

(i) [6 pts] $S_1, a_1, S_1, -16$. Fill in the Q-table after observing this transition and reward.

Q	State S_1	State S_2
Action a_1	-8	0
Action a_2	0	0

$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s) + \gamma max_{a'}[Q(s',a'))$
$Q(s_1,a_1) = (1-\alpha)0 + \alpha(-16 + \gamma max[0,0]) = -8$

(ii) [6 pts] $S_1, a_2, S_2, -20$. Fill in the Q-table with the updated Q values from part (i).

Q	State S_1	State S_2
Action a_1	-8	0
Action a_2	-10	0

$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s) + \gamma max_{a'}[Q(s',a'))$
$Q(s_1,a_2) = (1-\alpha)0 + \alpha(-20 + \gamma max[0,0]) = -10$

(iii) [6 pts] $S_2, a_1, S_1, +20$. Fill in the Q-table with the updated Q values from part (ii).

Q	State S_1	State S_2
Action a_1	-8	8
Action a_2	-10	0

$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s) + \gamma max_{a'}[Q(s',a'))$
$Q(s_2,a_1) = (1-\alpha)0 + \alpha(+20 + \gamma max[-8,-10]) = 8$

(iv) [6 pts] $S_1, a_2, S_2, -8$. Fill in the Q-table with the updated Q values from part (iii).

Q	State S_1	State S_2
Action a_1	-8	8
Action a_2	-7	0

$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s) + \gamma max_{a'}[Q(s',a'))$
$Q(s_1,a_2) = (1-\alpha)(-10) + \alpha(-8 + \gamma max[8,0]) = 8$

(b) [6 pts] What is the optimal policy for S_1 and S_2?

$\pi(s_1)$	$\pi(s_2)$
a_2	a_1

(c) Answer the following True/False questions.

(i) [3 pts] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.

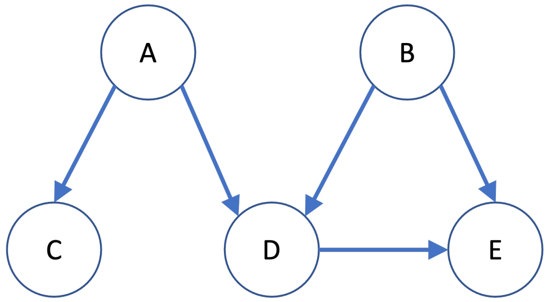
● True ○ False

(ii) [3 pts] Approximate Q-learning will always converge to the optimal policy.

○ True ● False

Q5. [39 pts] Bayes Nets and Sampling

Use the following Bayes Net for the questions below.



(a) [6 pts] List 3 conditional independences between variables A, B, C, and E. D should not be anywhere in the box below! If the variables are conditionally independent given nothing, use $X \perp\!\!\!\perp Y \mid \emptyset$

Independence 1	**Independence 2**	**Independence 3**
$A \perp\!\!\!\perp B, C \perp\!\!\!\perp B \mid A, C \perp\!\!\!\perp E \mid A$	$A \perp\!\!\!\perp B, C \perp\!\!\!\perp B \mid A, C \perp\!\!\!\perp E \mid A$	$A \perp\!\!\!\perp B, C \perp\!\!\!\perp B \mid A, C \perp\!\!\!\perp E \mid A$

(b) [3 pts] Which of the following is the expression for the numerator of $P(A \mid C)$?

○ $\sum_a P(a)P(B)P(C \mid a)P(D \mid a, B)P(E \mid B, D)$

○ $\sum_{a,c} P(a)P(B)P(c \mid a)P(D \mid a, B)P(E \mid B, D)$

● $\sum_{b,d,e} P(A)P(b)P(C \mid A)P(d \mid A, b)P(e \mid b, d)$

○ $\sum_{a,b,d,e} P(a)P(b)P(C \mid a)P(d \mid a, b)P(e \mid b, d)$

(c) [3 pts] For the numerator, if we want to reduce the size of the factor tables the most, what should the innermost (furthest right) summation be?

○ A ○ B ○ C ○ D ● E

(d) [3 pts] For the numerator, if we want to reduce the size of the factor tables the most, what should the outermost (furthest left) summation be?

○ A ● B ○ C ○ D ○ E

(e) [3 pts] Suppose you are computing factor tables for the denominator. How many factors (products of probabilities and summations) do we actually have to compute, not counting those that sum to 1? Do NOT count any remaining products outside of the outermost summation.

Factors to Compute:
1

(f) [8 pts] If we wanted to compute $P(A \mid c)$ using sampling, which methods would allow you to **directly** compute this quantity? Select all that apply.

☐ Prior Sampling
☒ Rejection Sampling
☐ Likelihood Weighted Sampling
☒ Gibbs Sampling

(g) [8 pts] In 1 sentence, what is one reason that would I want to use likelihood weighted sampling instead of prior sampling?

Reason.
Likelihood weighted sampling is used for cases where some of the variables are given, while prior sampling samples every variable irrespective of the given information.

Give an example of when likelihood weighting and prior sampling techniques may perform differently (i.e., different run time, computing different probabilities, etc).

Example.
If one of the variables that is given is of low likelihood, it may take prior sampling a long time to get enough samples of it, while LWS weighs the sample by the probability but doesn't wait until it is sampled.

(h) [5 pts] Suppose you want to compute the probability of $P(a, B, c, D, e)$ using likelihood weighted sampling.

Which variable(s) should you sample? Select all that apply.

☐ A ☒ B ☐ C ☒ D ☐ E ☐ None

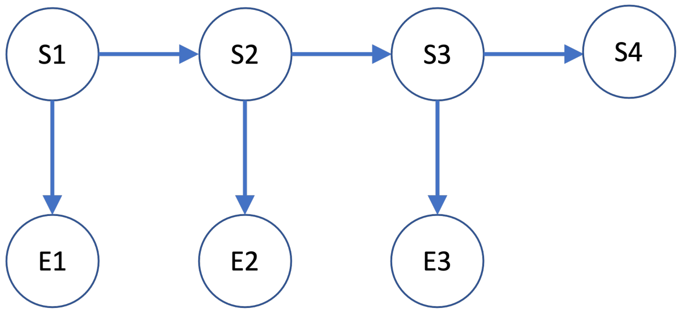
Which variable(s) contribute to the weight? Select all that apply.

☒ A ☐ B ☒ C ☐ D ☒ E ☐ None

Q6. [22 pts] Hidden Markov Models and Particle Filtering

Use the following HMM to answer the questions below.

Capital letters represent a table of all values of that variable. Lower case letters are variables specifying a single value (i.e., $S1 = s_1$). Note that two different variables may have different values (i.e., s_1 may not be equal to s_2), but if the same variable is used twice we mean that it has the same value.



(a) [18 pts] Given the HMM above and what we have learned about the structure of HMMs and their probability tables, which of the following must be equal? Select all that apply.

- ☐ $P(S2) = P(S3)$
- ☒ $P(S3 \mid S2) = P(S2 \mid S1)$
- ☐ $P(S3 \mid S2 = s_2) = P(S2 \mid S1 = s_1)$
- ☒ $P(E2 \mid S2) = P(E3 \mid S3)$
- ☐ $P(S2 \mid e_1, e_2) = P(S3 \mid e_1, e_2, e_3)$
- ☐ $P(S3 \mid e_1, e_2) = P(S3 \mid e_1, e_2, e_3)$
- ☐ $P(S4 \mid e_1, e_2, e_3) = P(S3 \mid e_1, e_2, e_3)$
- ☒ $P(S4 \mid e_1, e_2, e_3, s_2) = P(S4 \mid e_3, s_2)$
- ☒ $P(S4 \mid e_1, e_2, e_3, s_2) = P(S4 \mid e_2, e_3, s_2)$

(b) [4 pts] If the state changed based on an oracle dictating the decision, which of the following probabilities would be added to represent this new model? Select one.

○ $P(O_i|S_i)$ ● $P(S_i|O_i)$ ○ $P(O_i|E_i)$ ○ $P(E_i|S_i, O_i)$

(c) [2 pts] When would you prefer using exact inference with HMMs over particle filtering? Select all that apply.

- ☒ The state space is very small.
- ☐ The state space is very large.
- ☐ Solving runtime is more important than accuracy.
- ☒ Accuracy is more important than solving time.

(d) [2 pts] If we got rid of the resampling phase of particle filtering and instead just kept the particles we had where the weight of those particles is the product of the observation probabilities $P(e_i|S_i)$ that we've had up to and including the current time step, which of the following are true? Select all that apply.

- ☐ This will always work as well as standard particle filtering.
- ☐ This will generally work less well than standard particle filtering because the particles will cluster in the most likely part of the state space.
- ☒ This will generally work less well because the particles will tend to be in low-likelihood parts of the state space.
- ☐ This will generally work less well than standard particle filtering because the number of particles needed will increase over time.
- ☐ None of the above.

Q7. [24 pts] Game Theory

Fill in the following bimatrix games to satisfy the following constraints.

(a) [4 pts] (B,C) is the **only** Pure Strategy Nash Equilibrium. There should be no ties.

Player 2

		C	D
Player 1	A	1,	0,
	B	2,	3,

We chose (A,C) = (1,3), (B,C) = (2,2), (A,D) = (0,1), (B,D) = (3,0). B is dominant, so P1 will always choose B. Any solution that has $C > D$ for P2 is going to be the PSNE.

(b) [4 pts] D is a strictly dominant strategy for Player 2.

Player 2

		C	D
Player 1	A	1,	0,
	B	2,	3,

Any solution such that (A,C) < (A,D) and (B,C) < (B,D) is valid

(c) [8 pts] The utility of (A,C) for P2 is 3 and the utility of (B,D) for P2 is 4, and the strategies $s_1 = (1/2, 1/2)$ and $s_2 = (2/3, 1/3)$ are a mixed strategy Nash equilibrium. Find the utilities of (B,C) and (A,D).

Player 2

		C	D
Player 1	A	1 , 3	, 1
	B	2,	3 , 4

Using the values for P1 and strategy s_2, 1*2/3+D*1/3 = 2*2/3+3*1/3. When we solve for D, we get 5. Similarly, for P2's values and strategy s_1, we get 3*1/2+B*1/2 = 1*1/2+4*1/2. When we solve, we get B = 2.

(d) [8 pts] (Note the slightly different utilities) Player 1 is the leader with mixed strategy (0.5, 0.5), and C is Player 2's follower strategy, but there is no PSNE.

Player 1 \ Player 2	C	D
A	3,	0,
B	2,	3,

We need that 1) 0.5*[(A,C)+(B,C)] > 0.5*[(A,D)+(B,D)], 2) Player 2's utility for C ¡ D when Player 1 plays A (otherwise it would be PSNE), and similarly, 3) Player 2's utility for D < C when Player 1 chooses B. One such combination could be (A,C) = 2, (B,C) = 2, (A,D) = 3, (B,D) = 0.

THIS PAGE INTENTIONALLY LEFT BLANK