**CSci 384: Artificial Intelligence Spring, 2018**

Date: February 22nd (Thr.), 2018 Instructor: Dr. M. E. Kim

Due: 5:00 PM, March 2nd (Fri.), 2018 -- **No Extension of Deadline**

**Midterm**

Name: \_\_\_\_David Erickson\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Total: 300 point**

Please answer all of the questions at your best.

You have to show the sufficient steps and explanation for your answers. -- Sloppy answers will NOT get a full point.

Write your answers ***to this file*** and upload it to ‘Submission’ section at eZ-LMS: NO .pdf file is accepted.

Hours taken to complete the exam: \_\_\_ Hours \_\_\_ Minutes.

Mark the difficulty of the Exam:

Very Easy: \_\_\_\_\_ Easy: \_\_\_\_\_ Moderate: \_\_\_\_\_ Difficult: \_\_\_\_ Very Difficult: \_\_\_\_\_

Comment:

**Q1. [40] True/False. Explain/Justify your answer.**

For the following assertions, judge whether it is true or false. Justify your answers with an explanation, examples or counterexamples. Answer with no justification will get no point.

1. There exists a task environment in which no pure reflex agent can behave rationally.

**True. The card game memory is one. Anything where memory is required to do well will thwart a reflex agent.**

1. A perfectly rational poker-playing agent never loses.

**False. Pit two perfectly playing agents against each other, someone must lose.**

1. An admissible heuristic function h will never cause A\* to expand more nodes than the number of expanded by uniform-cost search.
2. Iterative deepening search will always find the same solution as depth-first search.

**False. DFS is not guaranteed to find an optimal path, Iterative deepening is. DFS may explore the entire graph before finding the target node, Iterative Deepening only does this if the distance between the start and end node is the maximum in the graph.**

1. Uniform cost search is a special case of A\* search with *h(n) = c* for any state *n.*

**False. A\* search: f(n)=g(n) + h(n)**

**Uniform-cost search: f(n) = g(n)**

1. In a game tree, α-β pruning algorithm will often produce better game-playing moves than the minimax algorithm.
2. Depth first search always expands at least as many nodes as A\* search with an admissible heuristic.

**False. Depth first search may sometimes expand fewer nodes than A\* with an admissible heuristic. But it is unlikely to happen without backtracking.**

1. Missionaries and cannibals problem is a constraint satisfaction problem so that we can solve it by a local search algorithm such as simulated annealing algorithm.

**True. Since there is a finite set of variables and moves, then it can be solved using a local search algorithm.**

**Q2. [30] Environment**

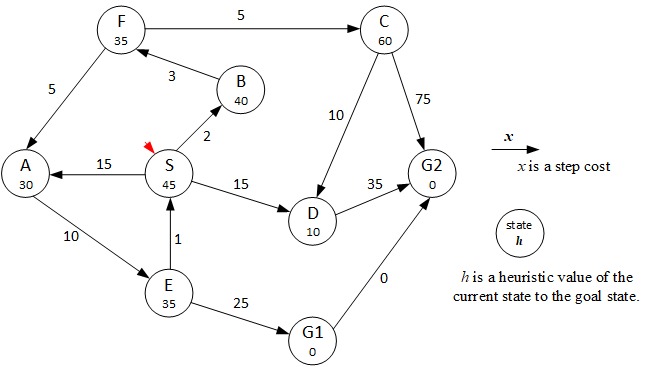
Give i) ***PEAS description*** of the task environment and ii) characterize it in terms of the ***properties*** of environment, e.g.) observable, deterministic, static, sequential, discrete, single-agent. Explain your answers.

1. Shopping for used AI books on the internet.
   * 1. **P: organized, fast, easy access, competitive prices E: internet, computer, money transfer A: internet, website, data transfer speeds, dollar amount S: keyboard, mouse, trackpad, touchscreen, UI**
     2. **Partial: only to see certain pages at a time, Deterministic: the current state depends on what youre shopping for and price range, sequential: your decisions on what book and other factors will determine where you will go, static: the websites will not change until the agent acts upon them, discrete: there are only so many websites and books that can be visited when shopping, multi-agent: there are many different agents that are active when browsing the internet.**
2. Playing a tennis against a wall.
   * 1. **P: hitting the ball, positioning, racket control E: player, tennis ball, racket, wall, ground A: players body, shoulder, wrist S: eyes, ears, touch**
     2. **Fully observable: you can see where that ball and player is and what they are doing always. Deterministic: the state of the ball changes depending on what the player does and how it hits the wall. Sequential: the players decisions will affect how the ball will be returned and how they will have to act in the future. Dynamic: the balls position will change while the player is deliberating what steps to do next. Continuous: there is a potential unlimited number of actions/reactions that could happen. Single Agent: Since the wall is acting as part of the environment, only the player can be counted as an agent.**
3. Playing soccer.
   * 1. **P: fast, agile, accurate, good teamwork, coordinated, verbal E: field, net, ball, players A: players legs, feet, head, goalie’s hands S: eyes, ears, touch**
     2. **Fully observable: you are able to see the position and movements of the ball and players at all times. Deterministic: the state of the game depends on all the previous actions that led them there. Sequential: the players next actions will determine the next state of the game. Dynamic: they state of the game will change while the players are deliberating what to do next. Continuous: there can be unlimited number of actions depending on the situation that the players are in. Multi-Agent: There are usually multiple players involved in playing soccer.**

**Q3. [80 pt.] Search Strategies**

Assume you have the following search graph, where S is the start node and G1 and G2 are goal nodes. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes (i.e. *h*-value).

For each of the search strategies listed below,(a) indicate which goal state is reached (if any), (b) list, in order, all the states expanded (i.e. in the explored list), and (c) give a solution as a sequence of states. (Recall that a (non-goal) state is expanded when it is removed from the frontier list and then added to the explored list.) When all else is equal, nodes will be expanded in alphabetical order. S is a start state and a goal test is applied when a node is removed from the frontier list.



1. [10] Breadth-First Search:

Goal state reached: \_\_G2\_\_\_

State expanded: \_\_\_A, B, D, E, F, G2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution: \_\_\_\_\_->S, D, G2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [10] Depth-First search:

Goal state reached: \_G1\_\_\_\_

State expanded: \_\_\_A, E, G1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution: \_\_\_\_\_->S, A, E, G1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [10] Uniform-Cost Search:

Goal state reached: \_G1\_\_\_\_

State expanded: \_\_\_A,D,B,F,A,C,E,D,G1\_\_\_\_\_\_\_\_

Solution: \_\_\_\_\_->S, B, F, A, E, G1\_\_\_\_\_\_\_

1. [10] Iterative Deepening Search:

Goal state reached: \_\_\_G2\_\_

State expanded: \_\_\_S,A,B,D,E,F,G2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution: \_\_\_\_->S,A,B,D,S,A,E,B,F,D,G2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [10] Hill Climbing Search (using function *h* only):

Goal state reached: \_\_\_\_\_

State expanded: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [10] A\* Search: a goal test is applied when it’s removed from the frontier list.

For the states expanded, show both the *states* and their *f values*.

Goal state reached: \_\_\_\_\_

State expanded: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [10] Imagine that the simulated annealing algorithm is applied. A node ‘F’ is the randomly chosen current node and a node ‘A’ is the candidate successor. Assume that the current temperature T equals 5.
2. What is the probability that node A will be accepted as the next state?
3. Assume that a node C is chosen as the candidate successor.

What is the probability that C will be accepted as the next state?

1. [10] The heuristic path algorithm is a best-first search in which the objective function is

*f(n) = (2−w)g(n) + w·h(n).* For what values of *w* is this algorithm guaranteed to be optimal? Explain your answer. You may assume that *h* is admissible.

**Q4. [20 pt.] Admissible Heuristic**

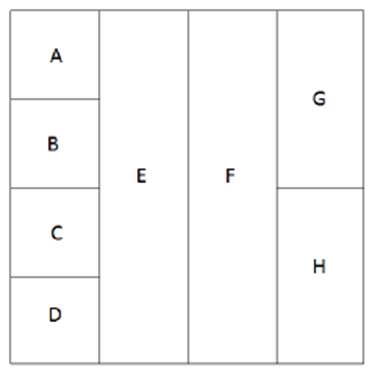
Assume you are given three admissible *h* functions for a given search problem; namely, *h1, h2*, and *h3*.

For each of the following combinations, explain whether or not the resulting *h* function is admissible.

**Q5. [35 pt.] Constraint Satisfaction Problem**

We plan to plant the flowers on April in the given land in the figure. The bulbs of the following flowers are ready to plant: daffodil, lily, tulip. Since each sub-land, A – G, belongs to a different owner, any adjacent sub-land should be distinguished by planting a different bulb.

1. [5] Formulate the given problem.
2. [5] Draw its constraint graph.
3. [10] Starting from the current partial assignment {A=daffodil, B = lily}, find a solution by HAND using the following strategy: backtracking search with MRV heuristic, the degree heuristic, least-constraining value heuristic, forward checking and/or constraint propagation (AC-3). At each step, you have to clearly specify what strategy is applied to improve the efficiency.

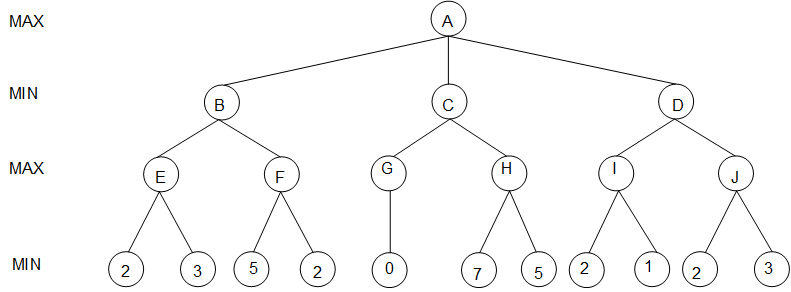


1. [5] If a constraint graph of problem can be reduced to a (nearly) tree-structured CSP, the problem may be solvable in linear time O(*n*) where *n* is the # of nodes. By conditioning method, decide a cycle cutset of the node(s) that need(s) to be instantiated.
2. [10] Using your cycle cutset in (4), solve it in the nearly tree-structured CSP (and/or) in the decomposed subproblems (if it’s a case). Show your solution step by step with the changes of domains.

**Q6. [35 pt.] The MiniMax algorithm and** **α-β** **Pruning**

Suppose that you’re a MAX player at the root A.

1. [10] (a) Decide in the ***minimax values*** at each node (A – J) in the Min-Max tree, and (b) ***mark the winning action*** for MAX player at the root A with a red line.

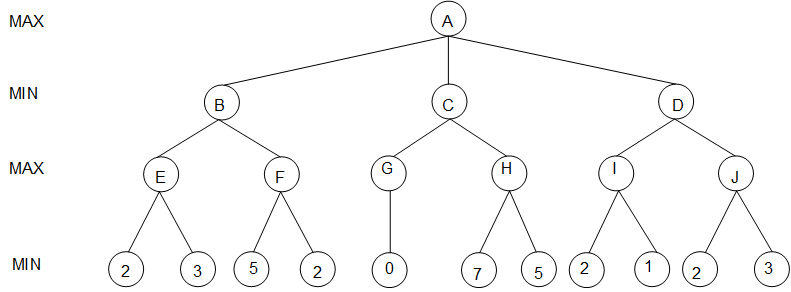


1. [15] Given the same Min-Max tree below,

(a) Show ***the changes of α or β values*** at each node,

(b) mark the ***branches that are pruned*** from the search by *α-β pruning*, and

(c) show the ***path*** chosen by MAX at the root.



1. [10] The effectiveness of *α-β pruning* search depends on the order of nodes in the Min-Max search tree. For the tree in (2), decides the best order and the worst order of nodes (B, C, D), respectively, and explain/justify your orders.

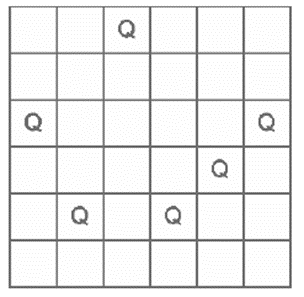
**Q7. [30 pt.] Simulated Annealing and Genetic Algorithm**

The *n*-queens problem requires you to place *n* queens on an *n* × *n* chessboard such that no queen attacks another queen. (A queen attacks any piece in the same row, column or diagonal.).

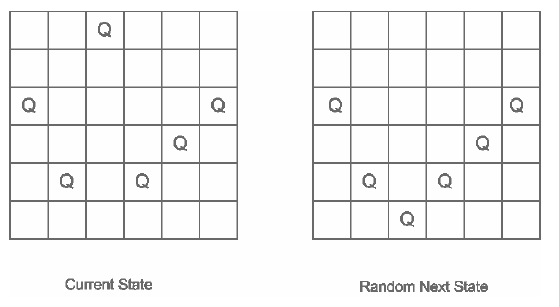
* We define the states to be any configuration where the *n* queens are on the board, one per column.
* The move set includes all possible states generated by moving a single queen to another square in the same column. The function to obtain these states is called the successor function.
* The evaluation function ***f*(*state*)** is the ***number of non-attacking pairs of queens*** in this state.

1. [5] How many possible states are there?
2. [5] For each state, how many successor states are there in the move set?

Suppose that a state is represented as a list of rows in which each queen is located from left. For instance, a state in the figure is (3, 5, 1, 5, 4, 3).



1. [5] What value will the evaluation function *f* (•) return for the above current state in the figure?
2. [5] If you use Simulated Annealing (currently T=3), and the current state and the random next state are shown below, will you accept this random next state immediately? or accept it with some probability? If it is the latter case, what is the probability?



1. [10] Suppose that you use a Genetic Algorithm. The current generation includes four states, S1 through S4. The evaluation values for each of the four states may be computed. Calculate the probability that each of them would be chosen in the selection step.
   1. S1 = (4, 6, 2, 6, 4, 4) Pr(S1) = \_\_\_\_\_
   2. S2 = (3, 5, 6, 5, 4, 3) Pr(S2) = \_\_\_\_\_
   3. S3 = (4, 6, 2, 4, 6, 2) Pr(S3) = \_\_\_\_\_
   4. S4 = (6, 5, 1, 3, 6, 6) Pr(S4) = \_\_\_\_\_
2. [10, optional] In (5), continue to solve the problem by applying the operations of genetic algorithm such as selection, crossover and mutation. Show the procedure of your solution step by step. The size of population is 4.

**Q8. [10] 3SAT and Hill-Climbing.**

SAT is the abbreviation for the satisfiability problem.

3SAT is the problem of finding a satisfying truth assignment for a sentence in a 3-CNF format, which is

defined as follows:

* A *literal* is a proposition symbol or its negation (e.g. *P* or *¬ P*).
* A *clause* is a disjunction of literals; a 3-clause is a disjunction of exactly 3 literals

(e.g. *P* ∨ *Q* ∨ *¬R*).

* A sentence in CNF or *conjunctive normal form* is a conjunction of clauses; a 3-CNF sentence is a conjunction of 3-clauses.

For example,

(*P* ∨ *Q* ∨ *¬ S*) ∧(*¬ P* ∨ *Q* ∨ *R*) ∧(*¬ P* ∨*¬ R* ∨ *¬ S*) ∧(*P* ∨*¬ S* ∨ *T*)

is a 3-CNF sentence with four clauses and five proposition symbols.

In formulating 3SAT problem, each state in the state space corresponds to an assignment of *True* or *False* to *every* propositional symbol. The successors of a state are defined as all those states which differ from the current state in the value of exactly one propositional symbol. The initial state is a random assignment. Assume that we use Hill-Climbing(without random restart) as the search strategy where the evaluation function used measures the number of satisfied clauses and may move to a neighbor with equal value.

1. [5] Given a 3-CNF sentence containing *n* distinct propositional symbols, how many successor states

should Hill-Climbing method evaluate from any given state? (Allow repeated states in the search tree.)

(2) [5] Consider the 3-CNF sentence:

(*¬ P* ∨ *Q* ∨ *R*) ∧(*P* ∨ *¬ Q* ∨ *R*) ∧(*P* ∨ *Q* ∨ *¬ R*) ∧(*P* ∨ *Q* ∨ *R*)

1. [5] Define a state (or states) that satisfy the goal condition.
2. [5, optional] What does the Hill-Climbing algorithm do next from the state defined by

(*P* = *False, Q* = *False, R* = *False*)?

**Q9. [20] Game Playing**

Consider the following game:

* When it is their turn to move, players must first choose which of two weighted coins,

A and B, to flip.

* Coin A comes up heads 10% of the time and tails the other 90%.

If heads, players must make move AH and if tails they must make move AT.

(To do this problem, you need not know exactly what each move means.)

* Coin B comes up heads 75% of the time and tails the other 25%.

If heads, the player must make move BH and if tails he or she (or it) must make move BT.

Assume it is the computers turn to play, and the game tree looks like the one below, where the values at the leaf nodes are the results of calls to the utility function, SBE (higher scores are better for the computer).

(1) [10] Explain what move the computer should make.

(Hint: think about expected-value calculations. Also, you might want to do parts it b and c first.)

1. [10] Now assume that there is no randomness and the players simply can choose any of the four moves (AH, AT, BH, or BT). Apply the minimax algorithm to the tree below and explain which move the computer should make. As in part (a), assume it is the computers turn to play.



**Q10. [30, optional] Cryptarithmetic Problem**

For the given cryptarithmetic problem,

CARROT + MINT + PEPPER = TOMATO

1. [10] Formulate the given problem.
2. [5] Draw its constraint graph.
3. [15] Solve the problem, using backtracking search with constraint propagation (AC-3), MRV heuristic, the degree heuristic, and/or least-constraining-value heuristic, etc. Show each step of your solution by specifying the domains of variables with the remaining values and specifying which heuristic/algorithm is applied to improve the efficiency.