KOÇ UNIVERSITY
College of Engineering

COMP 341: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Midterm 1

Fall 2021, 30/10/2021
Duration: 100 Minutes

Name: Solutions

ID: 00110001 00110000 00110000

- This exam contains 7 pages including this cover page and 4 questions. Check to see if any pages are missing. Put your initials on the top of every page, in case the pages become separated.
- By submitting this exam, you **agree** to fully comply with Koç University Student Code of Conduct, and accept any punishment in case of failure to comply. If you do not submit this exam on time, you will receive 0 credits.
- The exam is **closed book** and **closed notes**. You are **not** allowed to use any additional material including any electronic equipment such as computers and mobile phones.
- You are expected to be able provide clear and concise answers. Gibberish will not receive any credit. Your answers need to be readable by a human, illegible writing is not gradable.
- Read each question carefully and make sure to follow instructions. The written instructions take precedence over an answer that the instructor might give you during the exam, unless the instructor makes a class wide announcement.
- Do not write in the table below.

Question:	1	2	3	4	Total
Points:	24	38	22	16	100
Score:					

Honor Pledge

1. (24 points) True or False :

   True — The percepts of the agent affect how it chooses its actions.

   False — Pacman is a single-agent game.

   True — Planning agents require a model of the environment.

   True — Iterative Deepening Search always returns the same depth solution as Breadth First Search.

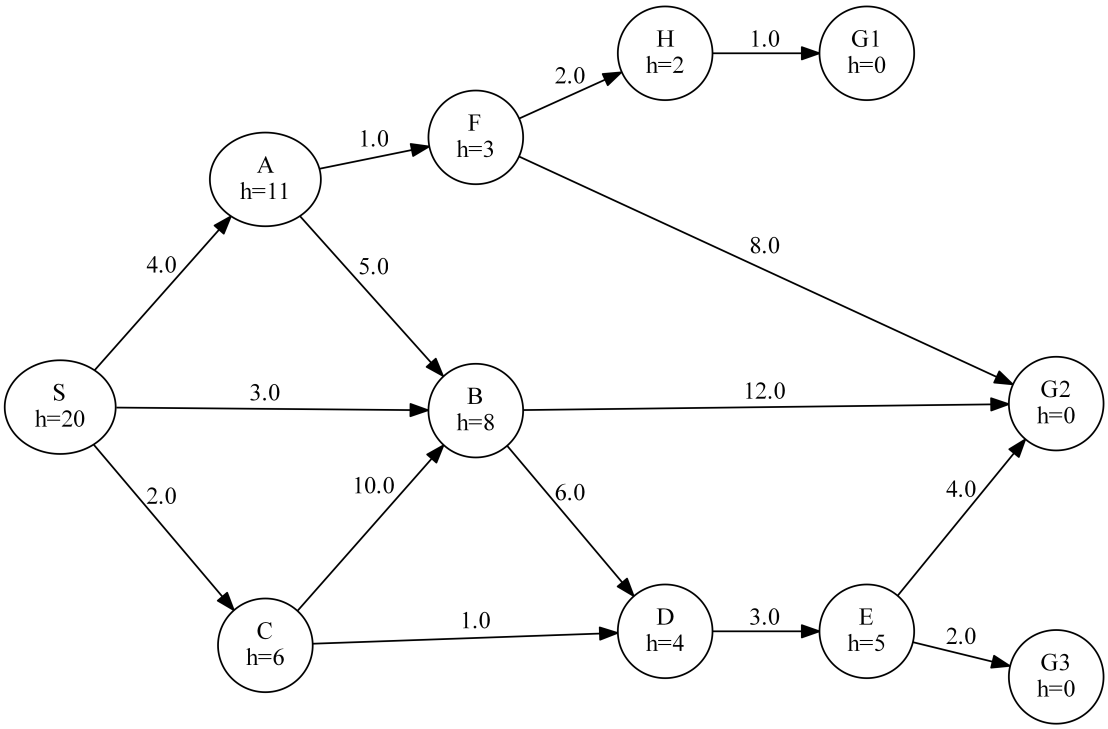
   True — Greedy Search is usually faster in finding a solution than A* search.

   False — In constraint satisfaction problems, no two variable can have the same value.

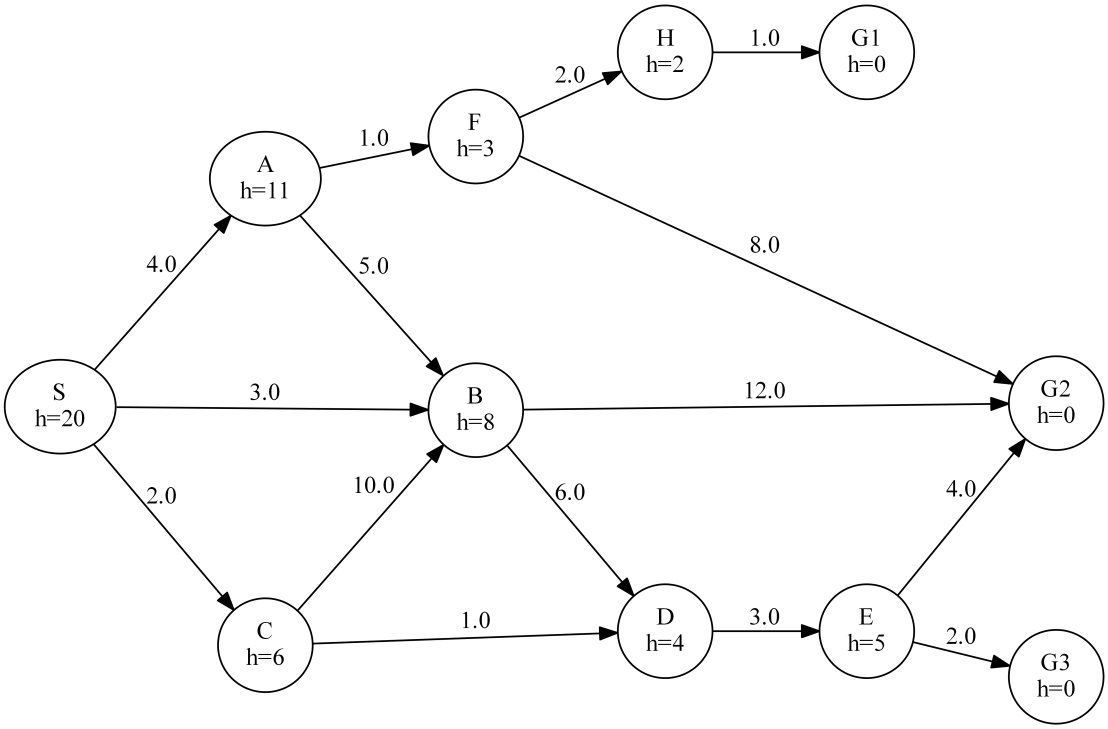
   False — A backtracking search does not always terminate.

   False — Genetic algorithms are prone to get stuck in local-minima.

2. (38 points) You are given the directed and weighted graph below, where **S** is the start state and **G1**, **G2** and **G3** are the goal states. The arcs represent transitions and cost of each transition is given next to the arcs. The numbers inside the nodes represent the heuristic value. For the given algorithms, write the popping from the frontier order of the nodes (Hint: S is the first and one of the goal states is the last) and the resulting solution path. Break any ties alphabetically (the alphabetical order is: A,B,C,D,E,F,G1,G2,G3,H,S). Use the graph search versions of the algorithms



(repeated to give you an additional figure to work on)



(a) (6 points) Breadth First Search:
Popped Node Order:
S,A,B,C,F,D,G2
Resulting Path:
S,B,G2

(b) (10 points) Uniform Cost Search:
Popped Node Order:
S,C,B,D,A,F,E,H,G1
Resulting Path:
S,A,F,H,G1

(c) (6 points) Greedy Search:
Popped Node Order:
S,C,D,E,G2
Resulting Path:
S,C,D,E,G2

(d) (10 points) A* Search:
Popped Node Order:
S,C,D,B,E,G3
Resulting Path:
S,C,D,E,G3

(e) (2 points) Is the given heuristic admissible?
No, look at A, cost is 4, heuristic is 11. This is why A* did not pick this

(f) (4 points) What is the optimal path from the start state to the lowest cost goal and what is the total cost of this path? There are actually two lowest cost paths! The one that UCS found and the one that A* found. Both of them have cost 8. Any selection will get you full points. If you write the two paths, we will give an additional point (total 5). If you argue why A* would never find the first one, you will get fruther bonus points

3. (22 points) We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to either land or take off. We have four time slots: $\{1; 2; 3; 4\}$ for each runway, during which we can schedule a landing or take off of a plane. We must find an assignment that meets the following constraints:

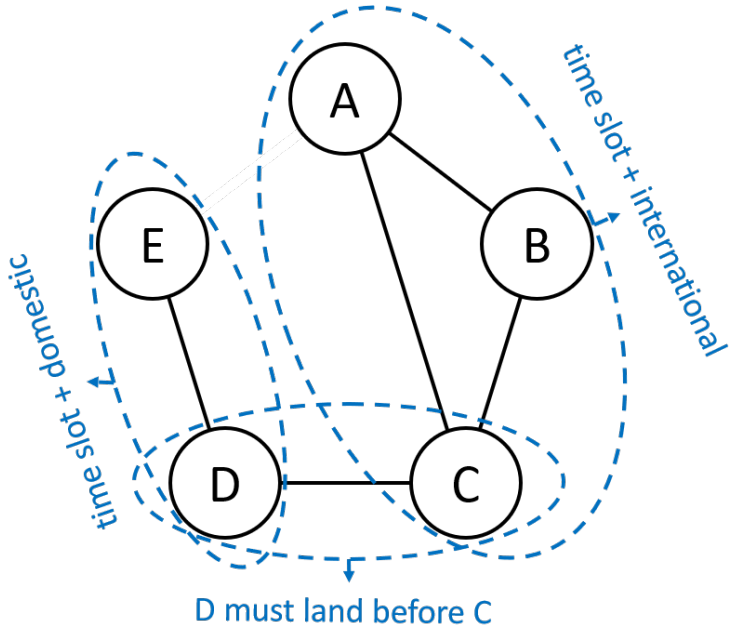
   - Plane B has lost an engine and must land in time slot 1.
   - Plane D can only arrive at the airport to land during or after time slot 3.
   - Plane A is running low on fuel and can last at most until time slot 2.
   - Plane D must land before C takes off, because some passengers must transfer from D to C.
   - Planes A, B, and C cater to international flights and can only use the international runway.
   - Planes D and E cater to domestic flights and can only use the domestic runway.
   - No two aircrafts can reserve the same time slot for the same runway.

   (a) (10 points) Formulate this as a CSP in terms of variables, domains and constraints. Express the constraints using mathematical or logical notation rather than with words. Draw the constraint graph.

   Variables are the planes: A,B,C,D,E (1 point)
   Domain is a tuple made up of the time slot and the runway: $(\{1, 2, 3, 4\}, \{domestic, international\})$ (2 points)
   Constraints: (0.75 point per correct constraint. -0.5 if a wrong constraint is given, +0.5 if all correct)

   - B[0] = 1
   - D[0] $\geq$ 3
   - A[0] $\leq$ 2
   - D[0] < C[0]
   - A[1] = B[1] = C[1] = *international*, D[1] = E[1] = *domestic*
   - A $\neq$ B $\neq$ C $\neq$ D $\neq$ E (or A[0] $\neq$ B[0] $\neq$ C[0] and D[0] $\neq$ E[0])

   Constraint Graph: (2 point)



(b) (6 points) What are the (partial-)domains of the variables after enforcing arc-consistency? (Cross out values that are no longer in the domain.)

A	~~1~~	2	~~3~~	~~4~~
B	1	~~2~~	~~3~~	~~4~~
C	~~1~~	~~2~~	~~3~~	4
D	~~1~~	~~2~~	3	~~4~~
E	1	2	~~3~~	4

A:1, C:1,2,3, D:4, E:3 are 0.75, others are 0.25. -0.25 for each mistake, 6.25 total

(c) (6 points) Arc-consistency can be rather expensive to enforce, and we believe that we can obtain faster solutions using only forward-checking on our variable assignments for this problem. Using the Minimum Remaining Values (MRV) heuristic, perform backtracking search on the graph, breaking ties by picking lower values and characters first. List the (variable; assignment) pairs in the order they occur (including the assignments that are reverted upon reaching a dead end). Enforce unary constraints before starting the search.

(B, 1, international), (A, 2, international), (C, 3, international), (C, 4, international), (D, 3, domestic), (E, 1, domestic):
Step 1: B has 1 remaining value as 1 which we pick. Forward checking (FC) removes 1 from A and C.
Step 2: A has 1 remaining value as 2 which we pick. FC removes 2 from C.
Step 3: C has 2 remaining values, 3 and 4. Pick 3 since it is lower. FC removes 3 and 4 from D, making it impossible solve. So we backtrack.
Step 4: Pick 4 for C. FC removes 4 from D.
Step 5: D has 1 remaining, 3 which we pick. FC removes 3 from E.
Step 6: Pick 1 for E since it is the lowest available. We are done.
All or nothing grading, make sure the assignments are correct.

4. (16 points) Local Search

   (a) (3 points) If we were to solve the CSP problem given in question 3 using a local search method, what would be a reasonable objective function? Be as clear as possible, using math and describing your notation is preferred.

   We perform local search with a fully specified state which may be violating the constraints. The number of violated constraints is a reasonable objective function.

   Let $c_i(s)$ represent the i^{th} constraint (ones we defined in Q3a) and we have n constraints. $c_i(s)$ is 1 if the constraint is violated in state s and 0 otherwise. Then the objective function is then $J(s) = \sum_{i=1}^{n} c_i(s)$.

   (b) (3 points) If we were to solve the CSP problem given in question 3 using a local search method, what would be your successor function?

   Basically we want to get from one state to another. There are multiple ways of doing this. Noting that each variable assignment is a two-tuple. Two immediate options.

   - Pick a single plane. Then change both tuples.
   - Pick a single plane. Then only change one of the tuples (time slot or the runway) in the odd iterations and the other one in the even. (Less preferred)

   Other reasonable answers will get points but changing everything randomly will not receive any points.

   (c) (6 points) Formulate the CSP problem given in question 3 to be solved by a genetic algorithm. Make sure to specify your state representation, fitness function and your successor function. You may copy your answer from a previous question, you do not have to change anything if it is not needed.

   Grading: State Representation 2, Fitness Function 2, Successor Function 2

   There are multiple answers to this. We want to be able to mix both the time slots and the runways. I will give two similar options.

   Option 1:

   - State Representation: $(t_A, r_A, t_B, r_B, t_C, r_C, t_D, r_D, t_E, r_E)$ where t_p is the time slot for plane p and r_p is the runway for plane p
   - Fitness Function: Prefferred $1/(1 + J(s))$ or less so $n - J(s)$ (since it does not put enough distance between the solution and few violations). n and $J(s)$ are from part a. Directly using $J(s)$ will get partial points.
   - Successor Function: Pick two individuals from the population of k with probability proportional to their fitness value, randomly select the crossover point, perfoms crossover and mutation steps. Do this k times.

   Option 2:

   - State Representation V1: Chromosome 1: $(t_A, t_B, t_C, t_D, t_E)$, Chromosome 2: $(r_A, r_B, r_C, r_D, r_E)$ where t_p is the time slot for plane p and r_p is the runway for plane p
   - Fitness Function: Prefferred $1/(1 + J(s))$ or less so $n - J(s)$ (since it does not put enough distance between the solution and few violations). n and $J(s)$ are from part a. Directly using $J(s)$ will get partial points (unless the selection is described accordingly)
   - Successor Function: Pick two individuals from the population of k with probability proportional to their fitness value, randomly select the crossover point, perfoms separate crossover and mutation steps for each chromosome. Do this k times.

   Notes: Almost all of you forgot the runway information. Instead of deducting points, I gave a bonus to those of you that added them.

(d) (4 points) Is it a good idea to apply genetic algorithms to the CSP problem given in question 3, based on your formulation? Why or why not? This is open ended and I will look at how you argued. We have some amount of unary constraint which will easily carry to the off-spring. From this perspective, the algorithm will work. We also have some binary constraints which tend to propagate. This makes the job of a genetic algorithm harder since multiple things must fit into place. Overall, it is not a bad idea but it is also not the best. From another perspective, this specific problem is very simple and there is no need for a genetic algorithm.