School of Computing and Information Systems

COMP90054 AI Planning for Autonomy

End of Semester 2 2017

Reading Time: 15 min

Writing Time: 120 min

This paper has 31 pages including this page and any Appendices.

Student Number:

Authorised Materials
Calculators, Drawing Instruments, Books, and (Lecture) Notes: Not permitted
Dictionaries: Paper based language translation dictionaries are permitted provided they are not annotated in any way.

Instructions to Invigilators:

- The examination paper is to remain in the examination room.
- Students will write all of their answers on this examination paper. Students may not remove any part of the examination paper from the examination room.
- Students should write their ID number on the examination paper.

Instructions to Students:

- This total marks for this paper is 50.
- Ensure your student number is written on the examination paper during writing time
- There are 5 questions. You should attempt all questions.
- You may use the reverse side of the page to make notes or prepare draft answers. The reverse sides will not be looked at or marked unless you clearly indicate that you wish this to be your final answer.
- You may use pencil.
- Mobile phones, tablets, laptops, and other electronic devices, wallets and purses must be placed beneath your desk.
- All electronic devices (including mobile phones and phone alarms) must be switched off and remain under your desk until you leave the examination venue. No items may be taken to the toilet.

Examiners use only:

Q1:	Q2:	Q3:	Q4:	Q5:

Question 1 **(12 Marks)**

This question concerns search.

(a). **[6 marks]** You are given a graph with 4 states: s_1 is the initial state and is connected to s_2 and s_3; s_2 is connected to s_4; and s_3 is connected to s_4. s_4 is the goal state. Is it possible to design a cost function $c(s)$ and heuristic function $h(s)$ such that A^* algorithm will not find an optimal solution? Justify your answer by showing the execution of the algorithm with the cost and heuristic function you've chosen. To show the working of the algorithm, show how the open and closed lists evolve after each node expansion.

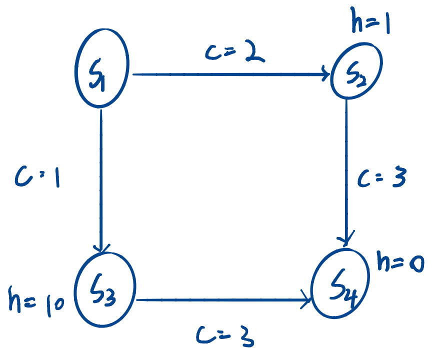


Figure 1

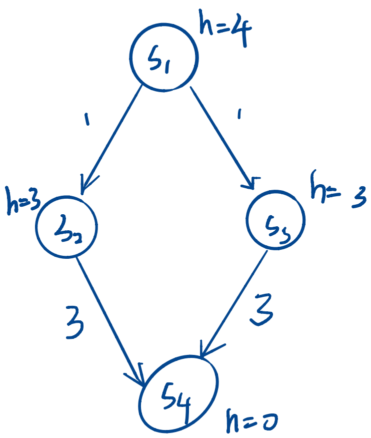
Possible: On Figure 1, c means cost, h means heuristic,

node structure : $\langle$ state, total cost, ancestor $\rangle$

Step	open	closed
1	$n_1 = \langle s_1, 0, - \rangle$	
2	$n_2 = \langle s_2, 2, n_1 \rangle$ $n_3 = \langle s_3, 1, n_1 \rangle$	$n_1 = \langle s_1, 0, - \rangle$
3	$n_4 = \langle s_4, 5, n_2 \rangle$ $n_3 = \langle s_3, 1, n_1 \rangle$	$n_1 = \langle s_1, 0, - \rangle$ $n_2 = \langle s_2, 2, n_1 \rangle$
4	$n_3 = \langle s_3, 1, n_1 \rangle$	$n_1 = \langle s_1, 0, - \rangle$ $n_2 = \langle s_2, 2, n_1 \rangle$ $n_4 = \langle s_4, 5, n_2 \rangle$

→ end

(b). **[6 marks]** Given an optimal heuristic h^*, can A^* **expand** more nodes than Hill Climbing with the same heuristic? Use an example to explain your answer using a graph of at most 4 nodes.



Yes, For A^*, it may generate 5 nodes in the example:

Suppose node structure: $n = <$state, totalcost, ancestor$>$

expand $n_1 = <S_1, 0, ->$, get $n_2 = <S_2, 1, S_1>$
$n_3 = <S_3, 1, S_1>$

expand $n_2 = <S_2, 1, S_1>$ get $n_4 = <S_4, 4, S_2>$

expand $n_3 = <S_3, 1, S_1>$ get $n_5 = <S_4, 4, S_3>$
END

which expand 3 times and get 5 nodes

For Hill climbing:

expand $n_1 = S_1, 0, ->$ get $n_2 = <S_2, 1, S_1>$
$n_3 = <S_3, 1, S_1>$

random break tie: remove n_3

expand $n_2 = <S_2, 1, S_1>$ get $n_4 = <S_4, 4, S_2>$
END

which expand 2 times and get 4 nodes

Question 2 **(13 Marks)**

This question concerns classical planning.

A robot can move between room A and room B. It has 1 hand that can pick-up or put-down 1 ball at a time. Every action has cost 1. Initially the robot starts at position A with 3 balls, and the goal is to have the 3 balls at room B. The robot can pick-up a ball if both are at the same location.

In answering the sub-questions below, you are allowed to use variables as arguments for the actions (action schemes), specifying the values of the variables. Note: it is **not** compulsory to use PDDL syntax, as long as you can convey the main ideas.

(a). **[6 marks]** Describe briefly in STRIPS how to model the actions 'move($from, to$)' and 'pickup($from, ball$)' and 'putdown($from, ball$)' using the predicates `robot(l)`, `ball(b,l)` `holding(b)` `and` `free` for each possible location l and ball b, where `robot(l)` `and` `ball(b,l)` stand for the robot and ball location, and `holding(b)` `and` `free` stand for the hand holding a ball and the hand when is free. You can use extra predicates if needed.

O = { move(from, to):

- Prec : robot(from)
- Add : robot(to)
- Del : robot(from)

pickup(from, ball):

- Pre : robot(from), free, ball(ball, from)
- Add: holding(ball)
- Del : ball(ball, from), free

putdown(from, ball):

- Pre : robot(from), holding(ball)
- Add: ball(ball, from), free
- Del : holding(ball)

}

(b). **[6 marks]** Model in STRIPS the Initial State s_0 and compute $h_{max}(s_0, G)$ and $h_{add}(s_0, G)$ for the goal G of the problem.

I = { robot(A), free, ball(Ball1, A), ball(Ball2, A), ball(Ball3, A) }

Suppose action cost is 1

hadd:

robot(A),	robot(B),	ball(Ball1, A),	ball(Ball2, A)	ball(Ball3, A)	ball(Ball1, B)	ball(Ball2, B)	ball(Ball3, B)	holding(Ball1)	holding(Ball2)	holding(Ball3)	free
0	∞	0	0	0	∞	∞	∞	∞	∞	∞	0
0	1	0	0	0	∞	∞	∞	1	1	1	0
0	1	0	0	0	3	3	3	1	1	1	0

$h^{add} = 9$

hmax

robot(A),	robot(B),	ball(Ball1, A),	ball(Ball2, A)	ball(Ball3, A)	ball(Ball1, B)	ball(Ball2, B)	ball(Ball3, B)	holding(Ball1)	holding(Ball2)	holding(Ball3)	free
0	∞	0	0	0	∞	∞	∞	∞	∞	∞	0
0	1	0	0	0	∞	∞	∞	1	1	1	0
0	1	0	0	0	2	2	2	1	1	1	0

$h^{max} = 2$

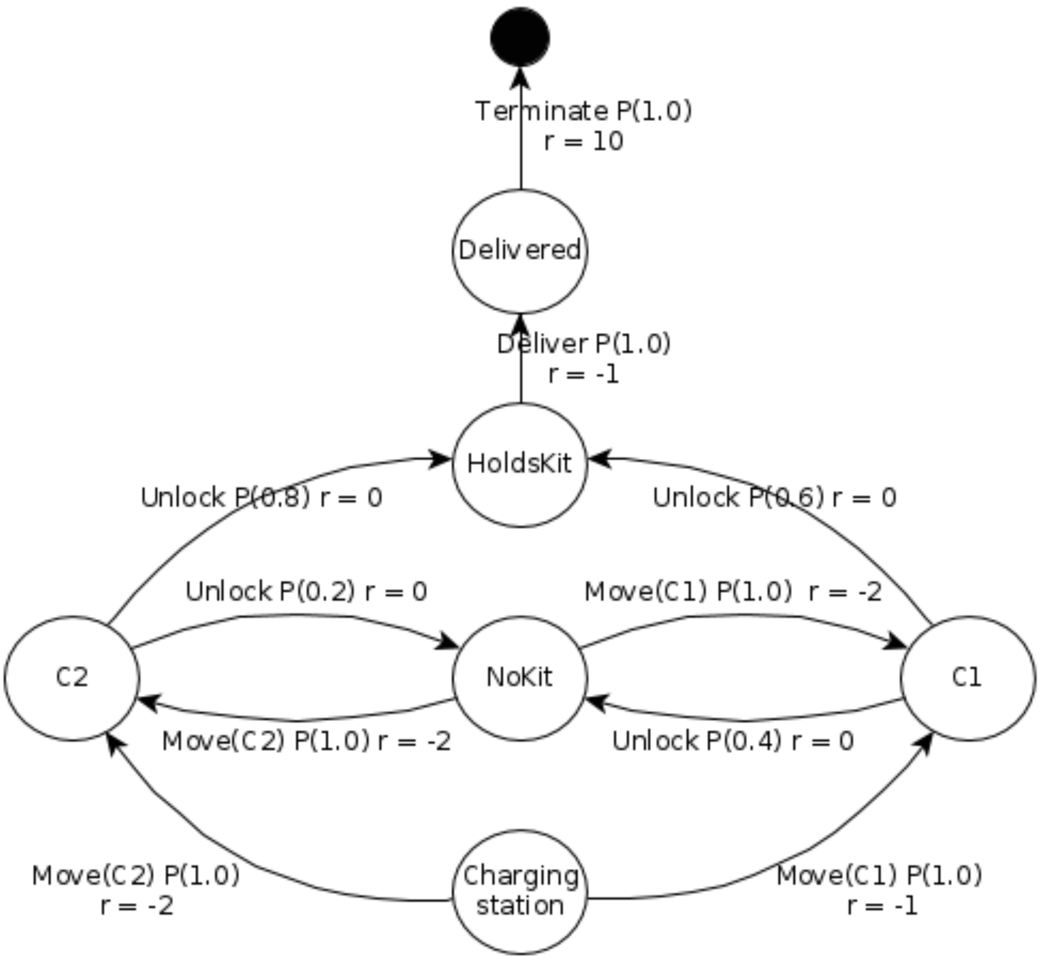
Question 3 **(12 Marks)**

Consider a robot called *MedAssist*, which is tasked with assisting medical staff in a hospital. Its primary task is to carry medical kits between rooms.

MedAssist is at its charging station fully charged and it receives a request to take a blood-testing kit to a doctor, who is in room 6.12. One blood-testing kit is in cupboard C1 and another is in cupboard C2. Cupboard C1 is closer to the charging station than cupboard C2. It would cost 1 unit (reward of -1) to move to cupboard C1 from the charging station and 1 unit to move from cupboard C1 to room 6.12. It would cost 2 units to move from the charging station to cupboard C2 and 1 unit to move from cupboard C2 to room 6.12. However, MedAssist assesses that there is a 60% chance that cupboard C1 will be locked, while only an 80% of cupboard C2 being locked. MedAssist is a bit stupid, so if it notes that a cupboard is locked, it will not updated these probabilities. Delivering the kit successfully to room 6.12 costs 1 unit, but this terminates the MDP, receiving a reward of 10.

??

The following diagram shows the transition probabilities, rewards, and values:



(a). **[3 marks]** Assume that MedAssist has calculated the following *non-optimal* value function V for this problem using value iteration with $\gamma = 1.0$. After 3 iterations it arrives at the following:

State	**Iteration** 1	2	3	4
V(Delivered)	10.0	10.0	10.0	
V(HoldsKit)	0.0	9.0	9.0	
V(NoKit)	0.0	0.0	0.0	
V(C1)	0.0	0.0	5.4	
V(C2)	0.0	0.0	7.2	
V(CS)	0.0	0.0	0.0	

If MedAssist is at the charging station (CS), which action should it choose to maximise its reward in the next state? Assume we are using the values for V after three iterations. Show full working to demonstrate that you understand the concepts.

$Q(CS, Move(C2)) = 1 \cdot (-2 + 7.2) = 5.2$

$Q(CS, Move(C1)) = 1 \cdot (-1 + 5.4) = 4.4$

$\because 5.2 > 4.4$

$\therefore$ it should choose Move(C2)

(b). **[5 marks]** Complete the values of these states for iteration 4 using value iteration. Fill the answers into the table, but show your working in the box. The first two have been completed already.

State	Iteration 1	2	3	4
V(Delivered)	10.0	10.0	10.0	10.0
V(HoldsKit)	0.0	9.0	9.0	9.0
V(NoKit)	0.0	0.0	why? 0.0	5.2
V(C1)	0.0	0.0	5.4	5.4
V(C2)	0.0	0.0	7.2	7.2
V(CS)	0.0	0.0	why? 0.0	5.2

$V(Nokit) = \max(Q(Nokit, Move(C2)), Q(Nokit, Move(C1)))$
$= \max(-2 + 5.4, -2 + 7.2) = 5.2$

$V(C_1) = Q(C_1, unlock) = 0.6(0+9) + 0.4(0+0)$
$= 5.4$

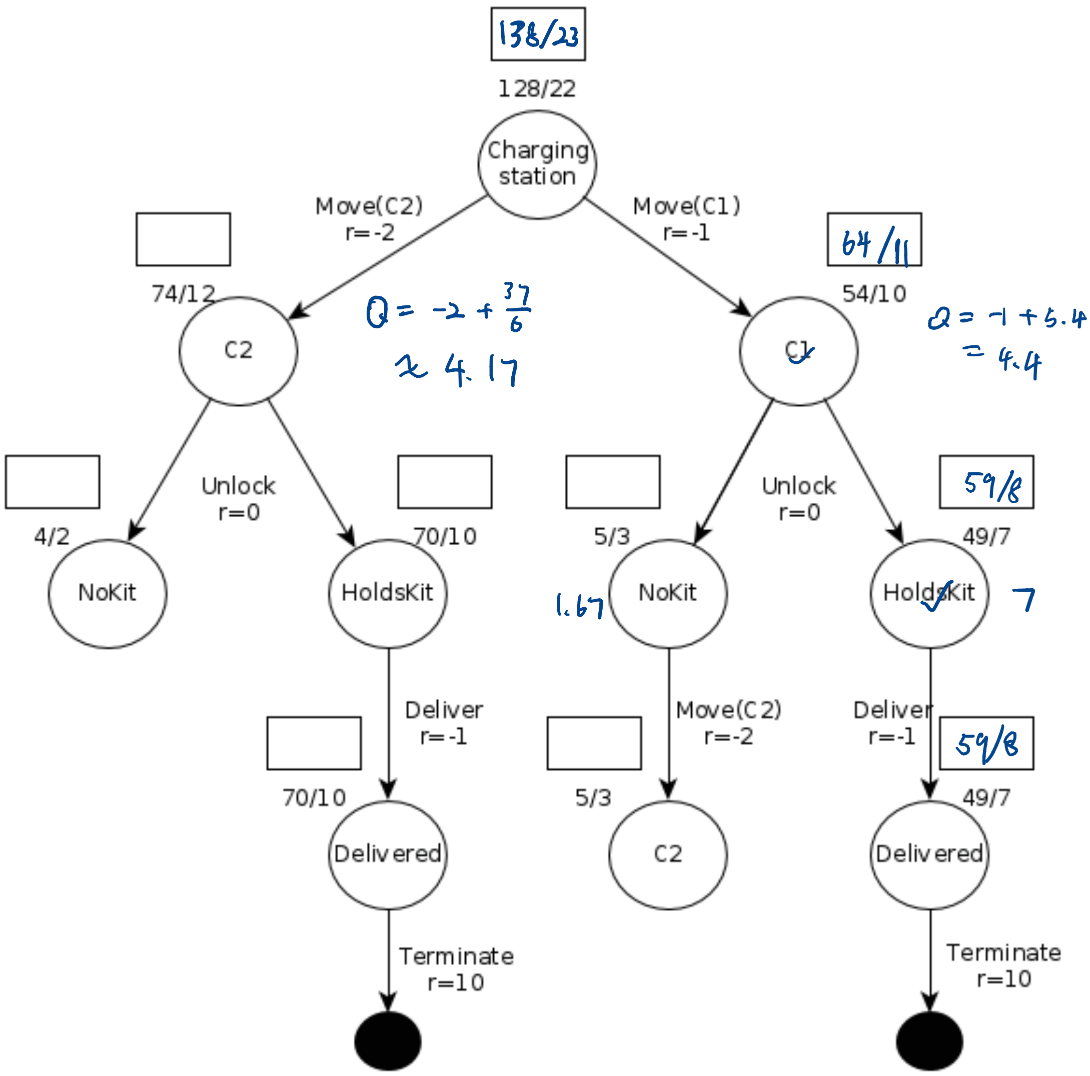
$V(C_2) = Q(C_2, unlock) = 0.8(0+9) + 0.2(0+0)$
$= 7.2$

$V(CS) = \max(Q(CS, Move(C_1)), Q(CS, Move(C_2)))$
$= \max(-2 + 7.2, -1 + 5.4)$
$= 5.2$

(c). **[4 marks]** You decide to implement a Monte-Carlo Tree Search (MCTS) approach for MedAssist to scale better. Below is a tree from MCTS in which 22 roll-outs have been performed. The black nodes represent end states (or terminal states). The score/payoff received for each roll-out is the cumulative reward for the path. That is, the value for any path that terminates is 10 (the reward for successful termination) plus the other rewards accumulated along the path (including negative rewards). The notation X/Y indicates that Y number of roll-outs have been performed, with a cumulative score of X, and thus X/Y is the average score and value for that state.

Using a greedy multi-armed bandit strategy (that is, ϵ-greedy with $\epsilon = 0$), and using the values on the tree as the base policy (that is, do not use the probabilities from the previous question), perform a complete iteration of the Monte-Carlo Tree Search algorithm. That is, perform *selection*, *expansion* (if required), and *back-propagation*. Note that simulation is not required: the selection/expansion should reach a terminal state.

To reach your answer, fill in the back-propagated values in the boxes on the diagram below. For those states not encountered during the iteration, leave the box empty.



Question 4 **(3 Marks)**

The hospital are so happy with the new MedAssist robot that they would like to order a second one. This requires two MedAssist robots to coordinate. On their first day, they receive requests for two kits, kit K and L, to be delivered to three different rooms R1, R2, and R3. They clearly cannot deliver two kits to three rooms simultaneously, so they need to each decide which rooms to go to. They cannot communicate.

Robot A is carrying kit K, while robot B is carrying kit L. Some of the instruments in kit K are also in kit L, so delivering both does not double the total reward.

The payoffs for each arriving in the different rooms are outlined below.

Circle all *pure strategy* equilibria of this game, if any exist.

Robot B (Kit L)

		R1	R2	R3
Robot A (Kit K)	R1	4, 4	~~5, 0~~	5, 5
	R2	5, 5	~~5, 0~~	5, 5
	R3	~~0, 5~~	~~0, 0~~	~~0, 5~~

Robot A will not choose R_3 (dominated by R_1 and R_2

Robot B will not choose R_2 (dominated by R_1, R_3)

∴ there are 3 pure strategy equilibria, that

is (R_1, R_3), (R_2, R_3), $(R2, R_1)$

the first action belongs to Robot A and

the second action belongs to Robot B

Question 5 **(10 Marks)**

The two MedAssist robots need are in a second scenario. Robot A has two kits, K and M, while Robot B has just one kit: L. Robot A is ordered to go to room R1 and Robot B to room R2. However, Robot R2 needs kits L and M. Robot A does not need kit M. The two robots need to determine whether to meet to swap kit M in the hallway between the two rooms. However, this will incur a cost.

If Robot A delivers kit K to room R1, it will receive a payoff of 7. However, to first meet in the hallway will cost 2, giving a payoff of 5.

If Robot B delivers only kit L to room R1, it will receive a payoff of 4. To first meet in the hallway will cost 2, however, if the robot first meets and then delivers kits L and M, it will receive a payoff of 6 (8 for delivering the kits minus the cost of 2).

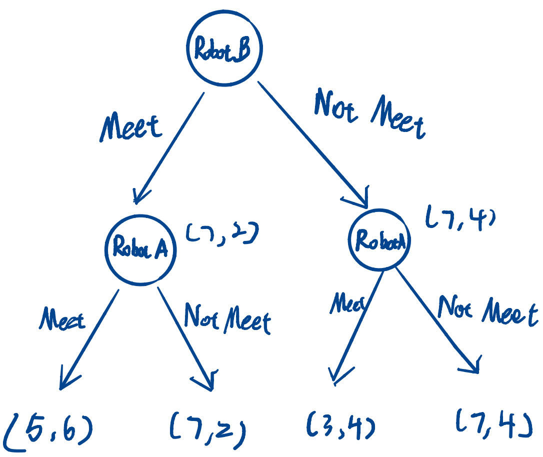
The normal form matrix for this is below.

		Robot B	
		Meet	Not meet
Robot A (kit K)	Meet	5, 6	3, 4
	Not meet	7, 2	7, 4

(a). **[5 marks]** Is there a unique Nash equilibrium in this game? Explain your working.

Yes, (Not meet, Not meet) = (7, 4) is the dominant strategy in this scenario

∴ unique Nash equilibrium: (Not Meet, Not Meet) in which the first action belongs to Robot A and the second action belong to Robot B (b). **[5 marks]** Assume that a new secure network is installed, allowing the MedAssist robots to communicate. Robot B decides that it will try to improve its utility by announcing its strategy ('Meet' or 'Not meet') to Robot A, allowing Robot A to subsequently choose its strategy. Assuming the payoffs remain the same as the previous question, which option should Robot B choose? Does this improve its utility? Show your working.



From the extended form tree, we can know that:

for Robot A, it will always choose Not Meet,

and for Robot B, it will better choose Not Meet

∴ Robot B should choose Not meet, this will not improve its utility.