COMP4620/8620 – Advanced Topics in AI: Intelligent Robotics

Semester-2 2025 – Assignment 2
Convenor & Lecturer: Hanna Kurniawati
Due date: Friday, 17 October 2025 23.59 Canberra time
Grace period: 12 hours after the due date (ie, 18 October 2025 00.00–11.59am Canberra time)
Late submission: NOT permitted —submission after the end of the grace period will automatically receive a mark of 0.

Please read the following notes first before starting to work on the assignment.

1. This is an individual assignment with a maximum total mark of 90 points and worth 35%.
2. This assignment consists of three parts: Part A, Part B, and in-person demo. The maximum mark for Part A is 30 points. The maximum mark for Part B is 30 points. The in-person demo has two components: (1) Demo of your program (question B3). (2) Q & A related to your submission. The maximum mark for component (2) of the in-person demo is 30 points.
3. Submission Instruction for Part A and Part B:
   (a) You must submit the following 2 files via wattle before the due date.
      i. One .pdf file containing answers to all the non-programming part of the assignment
      ii. One .zip/.7z/.tar.gz file containing your source codes and test cases.
   (b) For the Programming component, you must:
      i. Write your program in Python. We provide a set of functionalities you can use to help in this assignment. These functionalities are written in Python.
      ii. Submit all of your source codes and test cases used in your report. You must place all files (and folders) under a single top-level folder, compress the top-level folder into a single file with one of the following extensions: .zip or .7z or .tar.gz, and submit the compressed file. If your program consists of multiple files in multiple folders, your compressed file should preserve the folder structure. Note that during the in-person demo, you will download this file from Wattle, unzip, and run them in the lab's computer to solve the test scenarios we provide. Hence, all the files you need to run your program, aside from the libraries already installed in the lab's computer, must be included in this file.
   (c) Late submission is NOT permitted. We do provide a 12 hours grace period. Within the grace period, you can still submit and resubmit your assignment. However, after the grace period ends, you will NOT be able to submit your assignment. Without a submission, you will receive a 0 mark for this assignment.
4. Information for the in-person demo:
   (a) The in-person demo will take place on week-12. The exact date and location will be confirmed on Wednesday 24 September 2025.
   (b) **You must sign up for a demo slot at `https://bit.ly/comp4620A2DemoReg` before 3 October 2025 23:59 Canberra time**. The demo will be divided into 3 groups. To sign up for a demo slot, please check which group you are in, then in the sheet of your group, put your UID and name next to the demo slot you choose.
   (c) Please arrive at your demo place 10 minutes before the start of your demo slot and you must bring your ANU ID.
   (d) You will not be able to participate in the in-person demo if you do not submit Part A and Part B before the end of the grace period (see point 3 above).

Update:

- 24/09: Added a note in Part B Question 3.

PART A [30 pts]

1. [5 pts] Consider the vanilla value iteration algorithm (Algorithm 1). Will the lower bound on the number of iteration to converge to ϵ-close optimal solution be larger or smaller or the same if the discount factor is set to be larger? Please explain your answer.

Algorithm 1 ValueIteration($\langle S, A, T, R \rangle$)

1: Initialize $V^0(s) = R(s)$ for all state $s \in S$.
2: Let $t = 0$
3: **repeat**
4:     **for all** state $s \in S$ **do**
5:         $V^{t+1}(s) = \max_{a \in A} \left(R(s) + \gamma \sum_{s' \in S} T(s, a, s') V^t(s') \right)$
6:     Let $t = t + 1$
7: **until** $\max_{s \in S} \left| V^{t+1}(s) - V^t(s) \right| \leq 1e - 5$

2. [10 pts] In class, we have discussed that POMDP can be viewed as MDP in the belief space, which means Algorithm 1 can also be applied to a POMDP problem, given the belief space is discretized. Suppose we apply Algorithm 1 to a POMDP problem $P = \langle S_P, A_P, O_P, T_P, Z_P, R_P, \gamma \rangle$, where S_P, A_P, OP denote state, action and observation spaces respectively, while T_P, Z_P, R_P denote transition, observation and reward functions, respectively. Please answer the following questions and provide explanations to your answers.
   (a) [5 pts] What should we provide as the S, A, T, R input to Algorithm 1? If you need to do some pre-processing, please explain the pre-processing steps.
   (b) [5 pts] What is the time complexity of the inner loop (line 4–5 of Algorithm 1) in terms of the sizes of S_P, A_P, and O_P?

3. [7.5 pts] Consider an infinite horizon MDP framing of a robot navigating in a uniform grid environment, from a start cell s to a goal cell g. Suppose the robot can move one cell to the Left (L), Right (R), Up (U) or Down (D) of its current cell, or stay where it is. Each L/R/U/D move costs the robot -1, while staying where it is has 0 cost. Due to uncertainty in the effects of actions, the number of steps the robot needs to move from s to g vary between 8 and 15 steps. What should the reward for reaching the goal be, so that the optimal MDP policy enable the robot to reach g rather than staying where it is? Assume the discount factor $\gamma = 0.95$ and please explain your answer.

4. [7.5 pts] Congratulations, your start-up has just won a bid to develop a new fitness app for TibTif smartwatch. The app will be providing motivation to the smartwatch' user to keep an active lifestyle. A main feature of this app is its ability to account for the user's characteristics in providing the motivation. To simplify the problem, the app classifies the users' characteristics into three types: High, medium, and low motivation to keep an active lifestyle. For the highly motivated users, a gentle reminder is sufficient to keep them active. For the medium level of motivation, a warning on fatal diseases for lack of activity is required. Last but not least, for the users with low motivation, a very stern warning with some yelling is required.

   Of course, the users' characteristic is not easily assessable, and will never be known exactly. However, the smartwatch is equipped with a sensor that enables the smartwatch to keep track of the users' activity level. The sensor measures the level of activity of the user within one hour after the reminder/warning is given. It is known that a highly motivated user will be highly active during that one hour, while a user with medium level of motivation will be moderately active, and a user with low level of motivation will remain almost inactive. In this first version of the app, you can assume the user can always be active if they want to and ignore the fact that the user may not be able to be active due to commitment at the hour within the reminder/warning were provided.

Now, to keep the smartwatch price under A$200, the sensor in the smartwatch needs to be of a lower accuracy grade. It will only detect the right level of activity with 60% accuracy, with the rest of the probability mass equally distributed to the two wrong levels of activity the user has.

To enable the app to provide the right warning to the user (which might be different every time), your start-up decided to model the above problem as a POMDP. To this end, you need to provide the POMDP model to the engineering team, which is exactly your task in this question.

PART B [30 pts]

In this part, you are asked to perform a basic implementation, experiment, analysis and reporting that one generally does as part of a research work and paper writing in robot planning under uncertainty. However, of course the problem in this assignment has been significantly simplified to fit the assignment timeline.

Throughout this part of the assignment, consider the problem of centralised collision-free navigation of multi-drones in the presence of non-deterministic action effects, such as in the presence of wind disturbances. To simplify the problem, assume that the 3D environment is discretized into uniform grid cells of size $n_x \times n_y \times n_z$. Some of these cells are marked as obstacles. At each time step, each drone occupies a cell, and will make a decision of where to move next. A drone that occupies an obstacle cell is in collision, and two or more drones are in collision when they occupy the same cell. Collision will cause the mission to terminate. Each drone are initially located in one of the cells and has a goal cell to go to. Assuming that information about uncertainty of the effects of actions is provided as a probability function, how should the drones move, so that they can reach their goal cells as fast as possible without colliding with obstacles and each other? This part of the assignment explores MDP modelling of the above navigation problem and on-line solving of the MDP. To this end, you need to answer the following questions.

1. [3 pts] Define an MDP framing of the problem, assuming in a single step, a drone can move from a cell to one of the 26 neighbours of the cell and the uncertainty of the action effects are given as a probability function T_{known}.
2. [5 pts] Design a specific approximate on-line MDP solver and provide the pseudo-code. This means, if for instance you use MCTS, you need to specify the exact method you use for each of its four components.
3. [10 pts] Implement your pseudo-code, utilising primitive functionalities we provide (downloadable from `https://github.com/RDLLab/MultiDroneUnc` starting 26 September 2025). The mark for this question will come from the program's performance during the demo. Specifically, there will be 2 testing scenarios, ranging from a problem with only 1 drone that does not change altitude operating in a relatively open environment, to a problem with up to 4 drones that may change altitude operating in a cluttered environment with up to 5 obstacles. Your program has up to 2 minutes planning time per step with no parallelisation to solve a given problem. Each solved problem is worth 5 points.
Note 24/09: The test case we expect you to do well on will have less than 50 discrete actions, but we do have a challenge problem, which we can give if you are interested :)
4. [5 pts] Empirically investigate the performance of the MDP solver you proposed as the size of the action space increases. To this end, you can start by imposing a simplifying assumptions that all drones always fly at a certain altitude, and therefore, each drone can only move to its 8 neighbouring cells. You can increase the size of the action space by increasing the number of drones. You need to run systematic experiments on each scenario multiple times and report the summary statistics (at least mean and 95% confidence interval) of the experiments and discuss how the increasing size of action space affects the performance of your approximate solver.
5. [4 pts] Empirically investigate the performance of the MDP solver you proposed as the time per planning step increases. For this purpose, you need to use at least one of the scenarios proposed

in Question 4 and run them with increasing planning time per step. You need to run systematic experiments and report the summary statistics (at least mean and 95% confidence interval) of the experiments and discuss how the increasing time per planning step affects the performance of your approximate solver.

6. [3 pts] Discuss the limitation of your proposed solver.

Note that unlike Part A, the questions in this part are inter-related. For instance, your answer to question 3 must be an implementation of your answers to question 1 and 2, and your answers to questions 4–5 must be based on the results of running the code you implemented to answer question 3.

That's all folks