School of Information Technology, Deakin University

SIT320 — Advanced Algorithms

Distinction Task 12: **MDP and RL Algorithms**

About this Task and its related Module

At the completion of the module (**Module 12: MDL and RL Algorithms**), you are required to fill a lesson by doing following activities.

Your unit chair / tutor will then review your submission and will give you feedback. If your submission is incomplete they will ask you to include missing parts. They can also ask follow-up questions, either to clarify something, or to double check your understanding of certain concepts

Make sure your P and C tasks are either under-discussion or completed.

Task List

- **(0)** Provide a short overview of what you learned in the module. This should be based on your learning summary from lecture (seminar), module content on cloud Deakin, your interaction with Unit Chair/Tutors/Peers, your research in the library or the internet and/or your interaction with chatGPT (make sure to provide the prompts you use).
- **(1)** We discussed solving Tic-tac-toe using Minimax algorithm. You must have realised by now the limitations of Minimax (this should be reflected in your reflections). For this task, you are expected to modify the code for tic-tac-toe that you designed in Modules 1-3, and code another algorithm to solve it. You are expected to:
  - Formulate the problem as an MDP, that means, you will have a graph representing each board as a state. Note, you will not have access to transition probabilities, they will be initialised to random.
  - Code a solution to solve the MDP using value-iteration algorithm.
  - Devise and code a solution to tic-tac-toe using Q-Learning algorithm.
- **(2)** Test the working of your Q-Learning algorithm on Tic-tac-toe on board with size 5x5 or 7x7. To make your solution efficient, you are expected to code Monte-Carlo Tree Search algorithm and integrate it with Q-Learning algorithm that you coded in Task 1.
- **(3)** You are expected to record a short video ideally 5-8 minutes explaining your code along with salient features of the algorithm that you have implemented.