ECS/DSE-427/627: Multi-Agent Reinforcement Learning

Assignment-2

Question 1: **(30 marks)**

You are tasked with modelling and solving the Travelling Salesman Problem (TSP) as a reinforcement learning (RL) problem. The TSP involves an agent starting at a designated point and visiting a set of 50 targets, aiming to minimize the total cost of the journey.

The TSP code has been provided in the `marl-ecs-course` repository under `Assignment 2`.

Your task is to solve this problem using two different approaches:

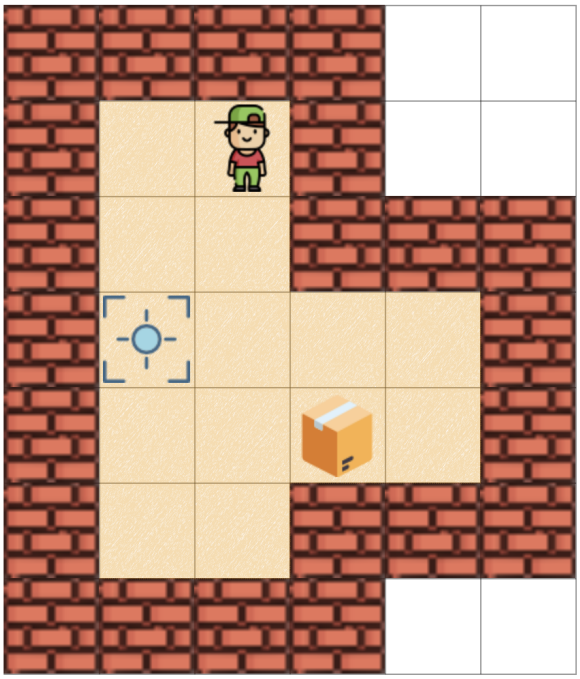
- Dynamic Programming (DP): Implement either value iteration or policy iteration.
- Monte Carlo (MC): Solve using Monte Carlo with exploring starts, comparing both the first-visit and every-visit methods.

Key Deliverables:

- Implement both algorithms (DP and MC) and compare their performances.
- Highlight and discuss the key differences between the two approaches based on your observations.

Question 2: **(30 marks)**

You are required to design and implement a grid-world-based environment for this assignment. The environment must follow the guidelines outlined below and replicate the dynamics of the classic Sokoban puzzle.



Environment Overview:

The game takes place on a grid of squares, where each square can either be a floor or a wall. Some of the floor squares contain boxes, while others are marked as storage locations. The objective is to guide a warehouse agent to push the boxes to their designated storage locations.

Key rules include:

- The agent can move horizontally or vertically onto empty squares (it cannot move through walls or boxes).
- The agent pushes boxes by walking up to them and moving them to the square directly beyond.
- Boxes cannot be pulled, and they cannot be pushed into walls or other boxes.
- The number of boxes equals the number of storage locations.
- The puzzle is solved once all boxes are placed at their respective storage locations.

For reference, you can check this visual example of Sokoban: Sokoban Animation.

Environment Details: (to be implemented in code, not provided)

1. **Grid Dimensions:**

   The grid world consists of a 6 x 7 board.
2. **State Space:**

   Each grid cell represents a state. For example, if the agent is located in row 2, column 3, the state is represented as (1, 2).
3. **Action Space:**
   a. The agent can move in four possible directions: UP, DOWN, LEFT, RIGHT. It can only push boxes forward and cannot pull them.
   b. Some actions are irreversible, such as pushing a box into a corner or the edge of the wall.
4. **Reward Structure:**
   a. The agent receives a reward of -1 if the box is not placed at the storage location.
   b. A reward of 0 is given when the box reaches its goal location.
5. **Termination Conditions:**
   a. The episode terminates when all boxes are successfully placed in storage locations.
   b. The environment also terminates if a box gets stuck in an unrecoverable position (e.g., pushed into a corner or against a wall).

Similar to Question-1, your task is to solve this problem using two different approaches:

- Dynamic Programming (DP): Implement either value iteration or policy iteration.
- Monte Carlo (MC): Solve using Monte Carlo with exploring starts, comparing both the first-visit and every-visit methods.

Key Deliverables:

- Implement both algorithms (DP and MC) and compare their performances.
- Highlight and discuss the key differences between the two approaches based on your observations.