CS747: Programming Assignment 2 Report

Name: Erata Maharshi
Roll No.: 210050049

18-03-2025

1. Task 1: MDP Planning

1.1 Implementation Details

- **Howard's Policy Iteration (HPI)**
  - *Policy Evaluation*: Solved using iterative policy evaluation with Bellman equations:
$$V_{k+1}(s) = \sum_a \pi(a|s) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$
  - *Policy Improvement*: Greedy action selection with first-max tie-breaking
  - *Terminal States*: Automatically set $V(s) = 0$ and excluded from policy updates
  - *Convergence*: Stopped when policy remained unchanged for 2 consecutive iterations
- **Linear Programming (LP)**
  - Formulated using PuLP (v2.4) with standard LP formulation:
$$\begin{aligned} \text{Minimize} \quad & \sum_{s \in S} V(s) \\ \text{Subject to} \quad & V(s) \geq \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V(s')] \\ & \forall s \in S, \forall a \in A \end{aligned}$$
  - Handled terminal states via explicit constraints $V(s) = 0$
- **Policy Evaluation Mode**
  - Solved $(I - \gamma P^{\pi})V = R^{\pi}$ using numpy.linalg.solve
  - Direct matrix inversion avoided for numerical stability

1.2 Design Decisions & Compliance

- **Default Algorithm**: Set HPI as default for better performance on small/medium MDPs
- **Floating-Point Handling**: Values printed with 6 decimal places using `f"%.6f"`
- **Input Parsing**:
  - States indexed 0 to $S-1$, actions 0 to $A-1$
  - Transitions are related with state, action pair to the following state, reward, probability tuple

1.3 Implementation

1.3.1 MDP File Parsing

- Processes MDP specification files, extracting states, actions, transitions, terminal states, MDP type, and discount factor
- Structures transitions as nested dictionary: `transitions[state][action] = [(next_state, reward, probability)]`

1.3.2 Policy Evaluation

- Solves Bellman equations for fixed policy using matrix algebra
- Handles terminal states by exclusion from equation system

1.3.3 Howard's Policy Iteration

- Iteratively evaluates policy via Bellman equations, then improves by maximizing Q-values
- Terminates when policy stabilizes (no action changes between iterations)

1.3.4 Linear Programming Formulation

- Formulates MDP as LP: $\min \sum V(s)$ s.t. $V(s) \geq \mathbb{E}[R + \gamma V(s')]$
- Derives policy by Q-value maximization post-solve

1.3.5 Main Execution Logic

- Parses command-line arguments for MDP file, algorithm choice, and optional policy evaluation
- Routes execution to HPI/LP solver or policy evaluation based on inputs
- Outputs results in required format: `value action` per state with 6 decimal precision

2. Task 2: Icy Gridworld

Not able to complete on time.