REPORT

ABHINAV N (20008)

Introduction

The assignment focused on two main tasks: designing a Markov Decision Process (MDP) for a student's activities on campus and solving a grid-world environment using Value Iteration and Policy Iteration techniques.

Question 1: Designing and Solving a Finite MDP

Task Overview: The first task involved designing a finite MDP for a student navigating between the hostel, academic building, and canteen on a university campus. The student could either attend classes or eat food at these locations, with different rewards and transition probabilities associated with each action.

Steps Taken:

1. **State Definition:**
   - Defined the possible states as locations combined with the student's action (e.g., attending classes, hungry).
   - The states were categorized into:
     - Hostel
     - Academic Building
     - Canteen

2. **Action Definition:**
   - Determined the possible actions:
     - Move to another location (hostel, academic building, canteen)
     - Stay in the current location

3. **Transition Probabilities:**
   - For each state-action pair, calculated the probability of transitioning to another state.
   - Example: From the hostel, the student had a 50% chance of going to the academic building and a 50% chance of staying in the hostel.

4. **Reward Structure:**
   - Assigned rewards based on the location:
     - Hostel: -1
     - Academic Building: +3
     - Canteen: +1

5. **Value Iteration:**

   - Implemented the Value Iteration algorithm to compute the optimal value for each state.
   - Iterated through states and actions to update the value function until convergence.

6. **Policy Iteration:**

   - Implemented the Policy Iteration algorithm to determine the optimal policy.
   - Alternated between policy evaluation and policy improvement until the policy stabilized.

7. **Results:**

1. **Optimal Policy Consistency**

Both Value Iteration and Policy Iteration yielded the same optimal policy:

- Classesπ(Hostel)=Attends Classes
- Building Classesπ(Academic Building)=Attends Classes
- Classesπ(Canteen)=Attends Classes

This consistency indicates that, regardless of the method used, the best course of action in each location is to attend classes. This makes sense given the reward structure, where attending classes in the Academic Building yields the highest reward (+3), compared to staying in the hostel or eating in the canteen.

2. Differences in Value Functions

While the policies are identical, the value functions differ significantly between the two methods:

- **Value Iteration Results:**
  - V(Hostel)=18.95
  - BuildingV(Academic Building)=20.94
  - V(Canteen)=19.81
- **Policy Iteration Results:**
  - V(Hostel)=13.10
  - BuildingV(Academic Building)=13.78
  - V(Canteen)=10.00

These differences reflect how each method approaches the problem:

- **Value Iteration:** The higher value functions indicate that Value Iteration was able to maximize the expected cumulative reward more effectively. Value Iteration updates the value of each state by considering the maximum possible reward obtainable from that state, leading to a more globally optimal solution.
- **Policy Iteration:** The lower value functions suggest that Policy Iteration may have converged to a different local optimum, where the expected rewards are lower. Policy Iteration alternates between evaluating the current policy and improving it, which can

sometimes lead to suboptimal value functions if the initial policy is far from the global optimum.

Question 2: Solving the 9x9 Grid-World Environment

Task Overview: The second task required solving a 9x9 grid-world problem, where an agent (robot) must navigate from a starting position to a goal, with the added complexity of tunnels acting as one-way portals.

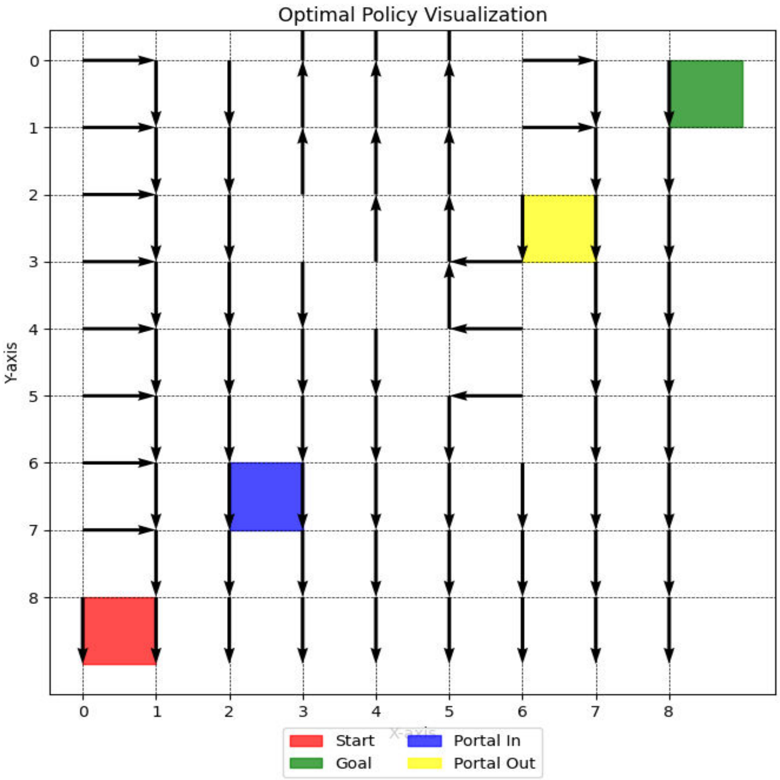
Steps Taken:

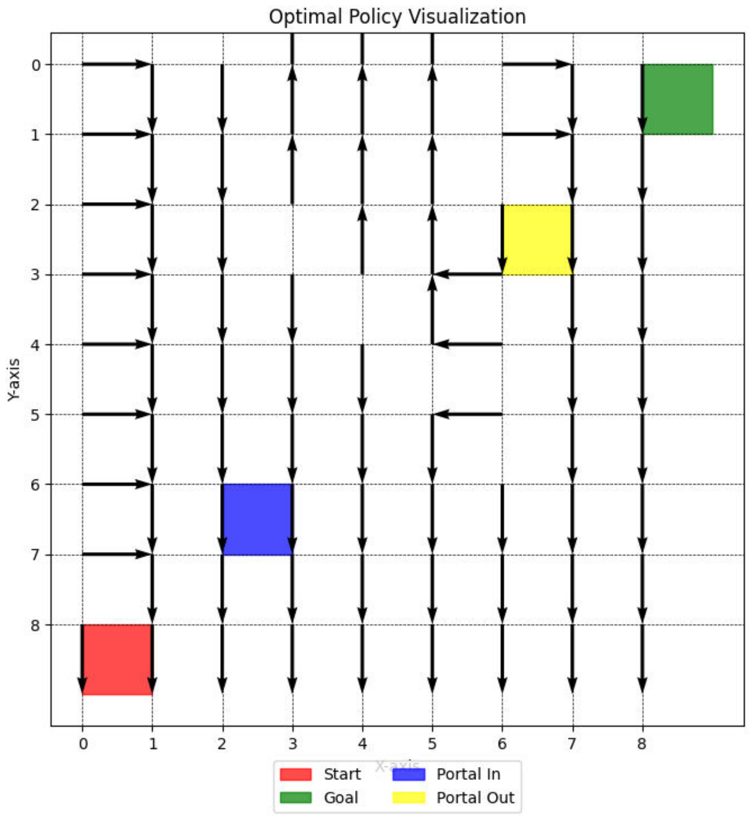
1. **Environment Setup:**
   - Defined the grid-world environment, including the robot's starting location, goal position, and the tunnels.
   - Assigned a reward of +1 for reaching the goal and 0 for all other states.

2. **Value Iteration Implementation:**
   - Applied Value Iteration to compute the optimal value for each grid cell.
   - Updated the value function iteratively for each state, considering possible actions and rewards until convergence.

3. **Policy Iteration Implementation:**
   - Applied Policy Iteration, starting with an arbitrary policy and refining it through policy evaluation and improvement steps until the optimal policy was found.

4. **Visualization:**
   - Compared the performance and results of Value Iteration and Policy Iteration.





Value Iteration vs. Policy Iteration:

- **Value Function:** Both Value Iteration and Policy Iteration produced identical value functions across the grid-world, indicating that both methods arrived at the same expected cumulative rewards for each state. The values gradually increase as the agent moves closer to the goal state, with the highest value (1.0) at the goal itself.
- **Optimal Policy:** Similarly, the optimal policies derived from both methods are identical, which shows that both techniques successfully identified the optimal movements for the agent in the grid-world environment. The agent is directed to move towards the goal using the shortest path, with clear instructions for moving "right" or "up" in most cases. The policy accounts for the one-way portals, as indicated by the transitions around those areas.