Unit-2

Lab experiments

Monte Carlo Control for Autonomous Drone Navigation

Aim:

An autonomous drone navigates a city grid to deliver packages. Using Monte Carlo control methods, implement a policy to optimize the drone's route to minimize delivery times and fuel consumption. Write the Python code to simulate this environment.

Algorithm:

1. **Initialize**
    - Define city grid size, start state, goal state, and obstacles
    - Initialize action set {Up, Down, Left, Right}
    - Initialize Q(s, a) arbitrarily for all state–action pairs
    - Set discount factor γ and exploration rate ε
2. **For each episode**
    - Set drone position to the start state
    - Initialize an empty episode list
3. **Generate an episode**
    - Select an action using **ε-greedy policy** from Q(s, a)
    - Execute the action and observe next state and reward
    - Store (state, action, reward) in the episode
    - Continue until the goal state is reached
4. **Compute returns**
    - Initialize return $G = 0$
    - Traverse the episode in reverse order
    - Update $G = \gamma G + \text{reward}$
5. **Update Q-values (First-Visit MC)**

- For each first-visited (state, action) pair in the episode
- Update Q(s, a) as the average of observed returns

6. **Policy Improvement**
    - Update policy $\pi(s) = \text{argmax}_a \, Q(s, a)$
7. **Repeat**
    - Repeat for many episodes until Q-values converge

Code Github Link:

https://github.com/syekumar/MLA0316-Reinforcement-learning-

output:

```
le   Edit   Shell   Debug   Options   Window   Help
Python 3.13.7 (tags/v3.13.7:bcee1
Enter "help" below or click "Help
>>>
========== RESTART: C:\Users\Dell
Optimal Policy (state : action):
(0, 0) -> 3
(0, 1) -> 1
(0, 2) -> 1
(0, 3) -> 3
(0, 4) -> 1
(1, 0) -> 3
(1, 1) -> 3
(1, 2) -> 3
(1, 3) -> 1
(1, 4) -> 1
(2, 0) -> 3
(2, 1) -> 0
(2, 3) -> 1
(2, 4) -> 1
(3, 0) -> 0
(3, 2) -> 3
(3, 3) -> 3
(3, 4) -> 1
(4, 0) -> 0
(4, 1) -> 2
(4, 2) -> 3
(4, 3) -> 3
>>>
```

Result:

- After training with Monte Carlo Control, the drone learns an optimal delivery policy.

- The learned policy guides the drone from Start (0,0) to Goal (4,4) using the shortest safe path.
- The drone avoids obstacles at (2,2) and (3,1) and does not hit grid boundaries.
- Delivery time is minimized by reducing the total number of steps.
- Fuel consumption is minimized due to step-wise fuel penalties.
- Q-values converge, indicating stable and optimal route selection.