Lab 11 planning with Markov Decision Process

1. Introduction

Markov Decision Process(MDP) consists of a tuple of 5 elements <s, A, p, r, γ>:

- s: Set of states. At each time step the state of the environment is an element $s \in S$.
- A: Set of actions. At each time step the agent choses an action $a \in A$ to perform.
- p: State transition model that describes how the environment state changes when the user performs an action a depending on the action a and the current state s.
- r: Reward model that describes the real-valued reward value that the agent receives from the environment after performing an action.
- γ: Discount factor that controls the importance of future rewards.

2. Planning with MDP

The goal of planning is to find a shortest path with collision avoidance. In this homework, we model the planning problem as MDP problem and use the value iteration or policy iteration algorithm to solve it.

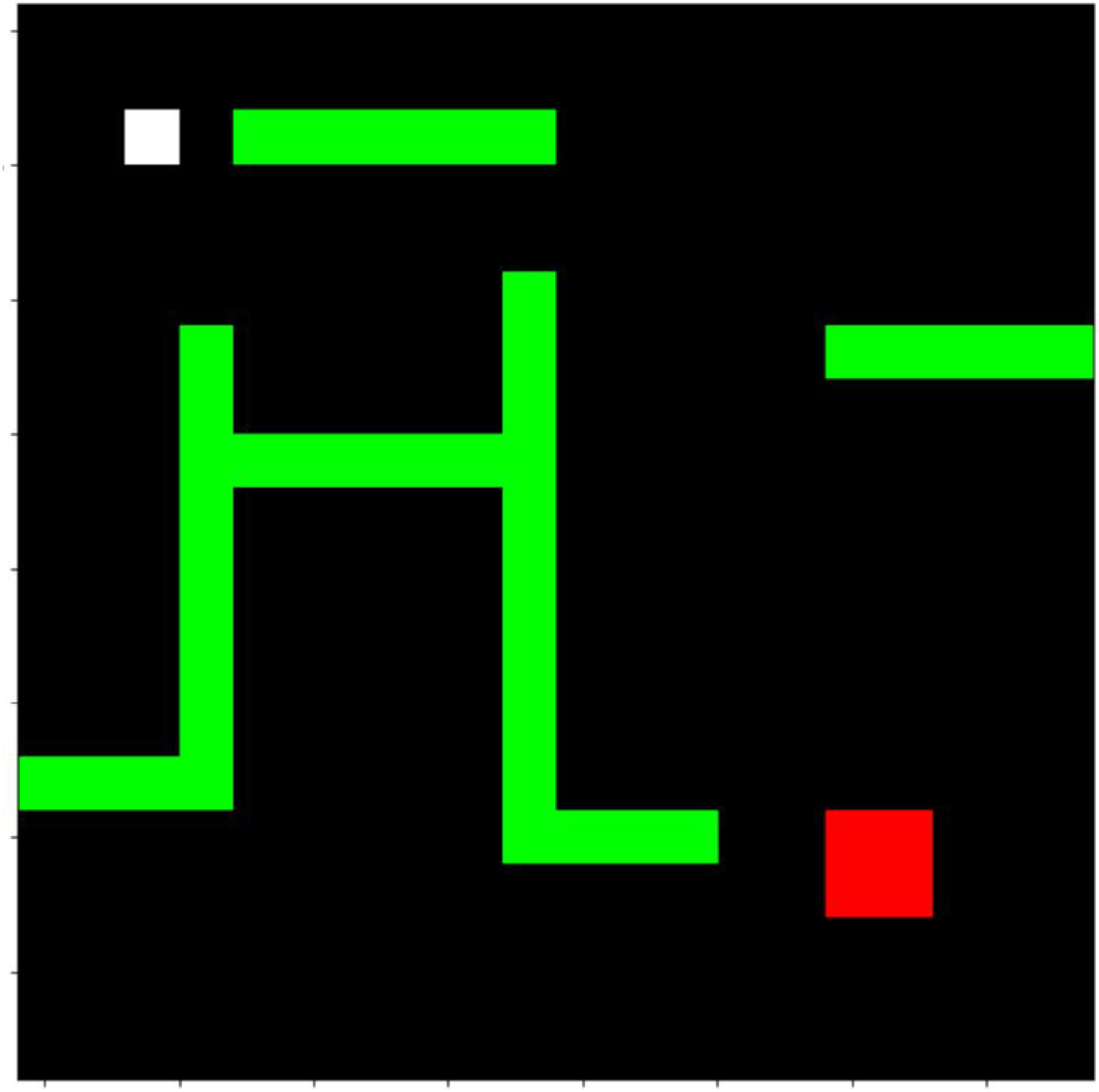
2.1 Homework Description:

Environment files:

- map_matrix.npy : the map matrix that show the environment. There are robot(white block), goal position(red block) and obstacles(green block), You can use the following code

```
import matplotlib.pyplot as plt
import numpy
map_matrix = np.load(“lab11_map_matrix.npy")
plt.imshow(map_matrix)
plt.show()
```

to show the color map (blue block):



To assign the white block for the path, you can refer to this code

> map_matrix[x, y, :] = 255

- reward_matrix.npy: The matrix of reward that demonstrates the reward at each state.

  goal position: 10; obstacle position: -10; other position: -1. To load this matrix (The reward provided by the lab might not be a good one, thus, you are encouraged to define the reward by yourself):

> import numpy
>
> reward_matrix = np.load(“lab11_reward_matrix.npy")

action and probability:

- left 0.25
- right 0.25
- up 0.25
- down 0.25

2.2 Requirement:

You should use value iteration or policy iteration algorithm mentioned in ppt to solve this problem.

2.3 evaluation:

metrics: 1. There is a path from start position to goal position with collision avoidance 2. Using the least steps.

Deadline: Submit your algorithm file and the image with the shortest path. 2019/12/3