Reinforcement Learning: Programming Exercise #2

In this programing assignment we are going to solve the problem explain in Figure 14.2 of the book "Probabilistic Robotics" using Dynamic Programing. (You don't really to refer to that book, everything needed is provided here)

The robot can move in 8 directions (4 straight + 4 diagonal). The robot has two model:

a- Deterministic model, that always executes movements perfectly.
b- Stochastic model, that has a 20% probability of moving +/-45degrees from the commanded move.

The world is given in the matrix below:

W = [

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1
1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1
1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

(1 means occupied and 0 means free).

The reward of hitting obstacle is -50.0 .

Reward for any other movement that does not end up at goal is -1.0.

The reward for reaching the goal is 100.0.

The goal location is at W(8,11) i.e. the red 0 in the matrix W above.

Use $\gamma = 0.95$.

You are required to generate the optimal policy for the robot using the following algorithms:

1- Policy iteration (algorithm on page 80 Sutton and Barto)
2- Value Iteration (algorithm on page 83 Sutton and Barto)
3- Generalized Policy Iteration

Monitor the convergence rate of each algorithm, which one converges faster?

You should be generating plots like the ones below for optimal policy (for both robot model “a” and “b” explained above). The optimal policy for the stochastic robot avoids narrow passages and tries to move to the center of corridors.

You plots should look like figure (a) for deterministic motion model and figure (b) for stochastic motion model.

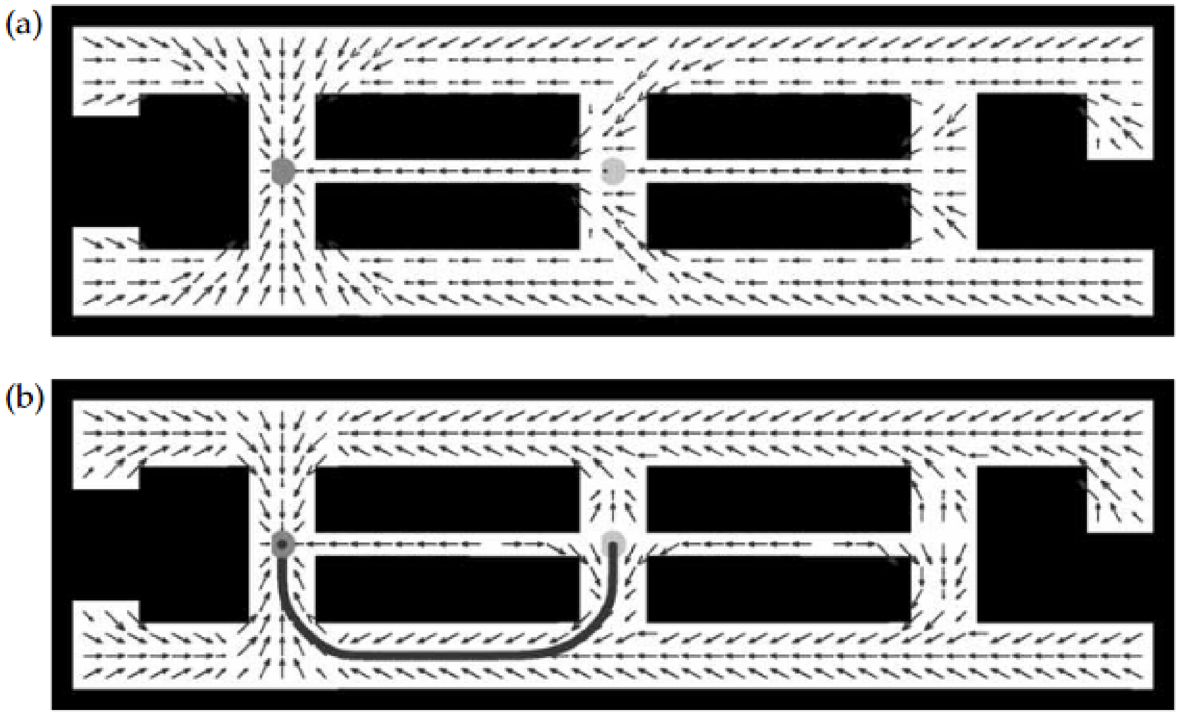

Figure 14.2 The value function and control policy for an MDP with (a) deterministic and (b) nondeterministic action effects. Under the deterministic model, the robot is perfectly fine to navigate through the narrow path; it prefers the longer path when action outcomes are uncertain, to reduce the risk of colliding with a wall. Panel (b) also shows a path.

In addition, you need to plot the value function in a plot like this: