

Assignment 6 Report for Part A

1. Using the menu commands, set up the MDP for TOH with 3 disks, no noise, one goal, and living reward=0. The agent will use discount factor 1. From the Value Iteration menu select "Show state values (V) from VI", and then select "Reset state values (V) and Q values for VI to 0".

Use the menu command "1 step of VI" as many times as needed to answer these questions:

1a. How many iterations of VI are required to turn 1/3 of the states green? (i.e., get their expected utility values to 100).

4 iterations of VI are required.

1b. How many iterations of VI are required to get all the states, including the start state, to 100?

8 iterations of VI are required.

1c. From the Value Iteration menu, select "Show Policy from VI". (The policy at each state is indicated by the outgoing red arrowhead. If the suggested action is illegal, there could still be a legal state transition due to noise, but the action could also result in no change of state.) Describe this policy. Is it a good policy? Explain.

Most of the policies are move to the right. However, in some case, this policy is illegal. It is obviously a bad policy, because it can not show the path to the goal state. The main reason that makes this situation is because the discount factor is 1. No matter what the number of steps is, the states value would be same. That is why it cannot show the path.

2. Repeat the above setup except for 20% noise.

2a. How many iterations are required for the start state to receive a nonzero value.

8 iterations of VI are required.

2c. At this point, view the policy from VI as before. Is it a good policy? Explain.

Move a disk from Peg 1 to Peg 3. It is a good policy.

2d. Run additional VI steps to find out how many iterations are required for VI to converge. How many is it?

25 iterations of VI are required.

2e. After convergence, examine the computed best policy once again. Has it changed? If so, how? If not, why not? Explain.

It does not change. Because 8 iterations are enough to generate a good policy, and more iterations just confirmed it.

3. Repeat the above setup, including 20% noise but with 2 goals and discount = 0.5.

3a. Run Value Iteration until convergence. What does the policy indicate? What value does the start state have? (start state value should be 0.82)

The policy indicates the final state would be the smaller reward state, and the start state is 0.82.

3b. Reset the values to 0, change the discount to 0.9 and rerun Value Iteration until convergence. What does the policy indicate now? What value does the start state have? (start state value should be 36.9)

The policy indicates the final state would be the bigger reward state, and the start state is 36.9.

4. Now try simulating the agent following the computed policy. Using the "VI Agent" menu, select "Reset state to s0". Then select "Perform 10 actions". The software should show the motion of the agent taking the actions shown in the policy. Since the current setup has 20% noise, you may see the agent deviate from the implied plan. Run this simulation 10 times, observing the agent closely.

4a. In how many of these simulation runs did the agent ever go off the plan?

The agent went off the plan for only one time.

4b. In how many of these simulation runs did the agent arrive in the goal state (at the end of the golden path)?

The agent arrived the goal state for 9 times.

4c. For each run in which the agent did not make it to the goal in 10 steps, how many steps away from the goal was it?

It was 2 steps away from the goal.

4d. Are there parts of the state space that seemed never to be visited by the agent? If so, where (roughly)?

The 'top triangle' never be visited by the agent.

5. Overall reflections.

5a. Since it is having a good policy that is most important to the agent, is it essential that the values of the states have converged?

It should be converged so that the optimal path can be generate, or the optimal path would always change if the number of iteration changes.

5b. If the agent were to have to learn the values of states by exploring the space, rather than computing with the Value Iteration algorithm, and if getting accurate values requires re-visiting states a lot, how important would it be that all states be visited a lot?

All states should be visited to generate a descent path. Because the value of every state should be compared so that the optimal path can be guaranteed.