CS6700 : Reinforcement Learning
Programming Assignment Report-1
Control Algorithms

Name: Pragnesh Rana **Roll number:** ME17S301

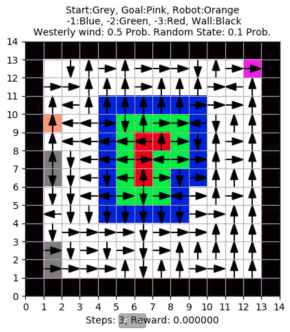
- Part-A is on Puddle GridWorld
- Part-B is about Policy Gradient Implementation

newpage

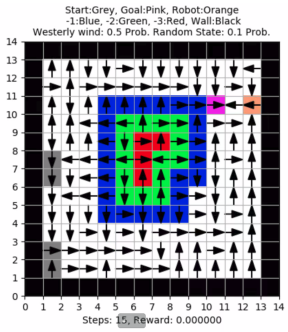
1. Puddle World:

Puddle world has been implemented with different maps. Map-A and Map-B has different terminal states whereas, Map-C has westerly wind blowing which forces the agent to move in east with 0.5 probability. The grid world has puddle in the centre. each state in the world is denoted by different colours.

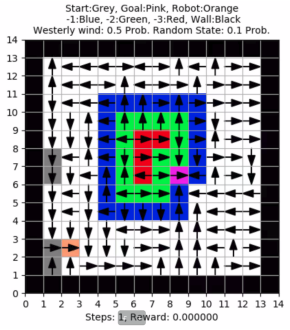
Penalty/reward given to the agent varies based on state and transition. For every transition from one state to another 0 reward is given. In the puddle zone, blue area gives reward of -1 likewise as inside penalty increases. Green and red gives -2, -3 respectively. Puddle world is stochastic in nature. It takes correct action with probability 0.9 where as other action are taken with 0.033 probability equally.Pink shows the terminal state which gives reward of +10 as shown in fig:1.



(a) Puddle World-A



(b) Puddle World-B



(c) Puddle World-C

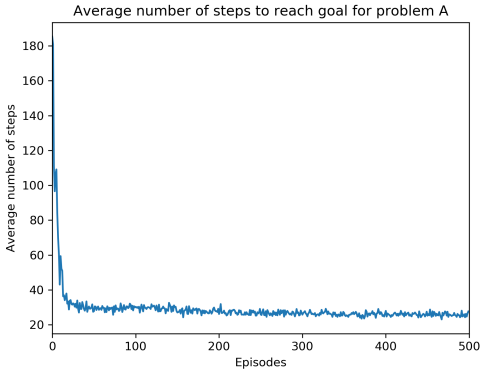
Figure 1: Puddle World with different colour: Grey:Start, Pink:Terminal, Middle:Puddle

In Puddle World-C, westerly wind also blowing which forces the agent to move in east direction with probability of 0.5. Suppose agent wanted to go in desired state s' with probability 0.9 and reward will be -1 due to transition but due to wind it may happen that agent may end up in the another state s" with probability 0.5 and assume such state has higher negative reward then wind misguides the agent.

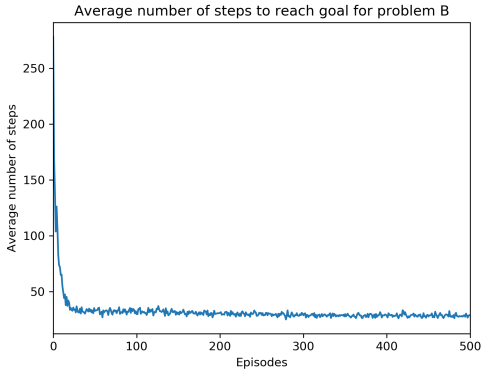
1.1. Q-learning:

The goal is to reach the terminal state with highest possible reward. ran the code for 500 episodes and average steps and return has been computed for 50 runs. For each case discount rate γ is taken as 0.9, learning rare α is taken as 0.1 and exploration parameter is taken as ε= 0.1.

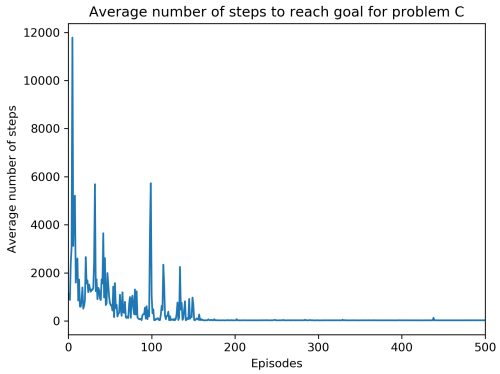
Epsilon-greedy policy has been used for the exploration. Initially the epsilon was set as 0.1 but as it takes may steps to compute the terminal state. so epsilon value has been boosted to 0.2 .



(a) Puddle World-A

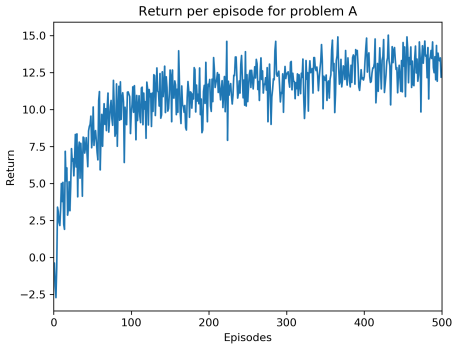


(b) Puddle World-B

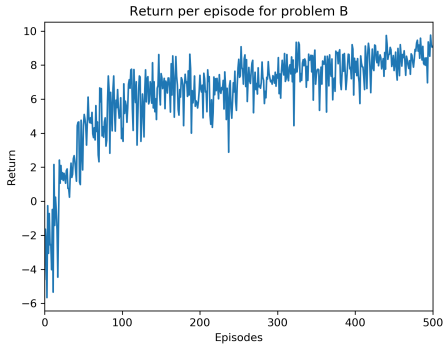


(c) Puddle World-C

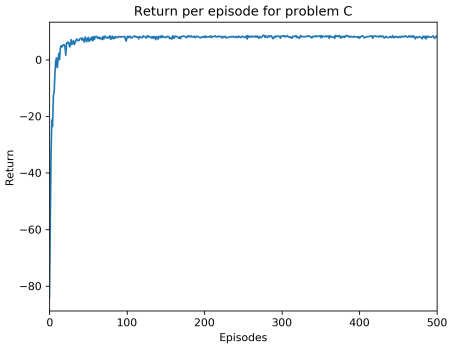
Figure 2: Average steps taken by the agent in different world



(a) Puddle World-A



(b) Puddle World-B

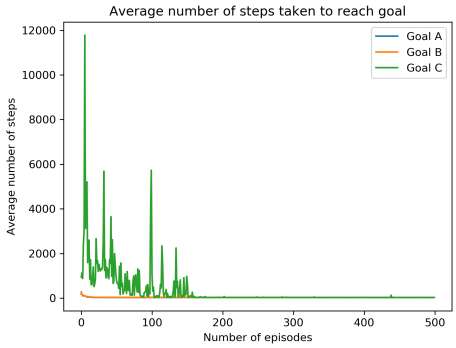


(c) Puddle World-C

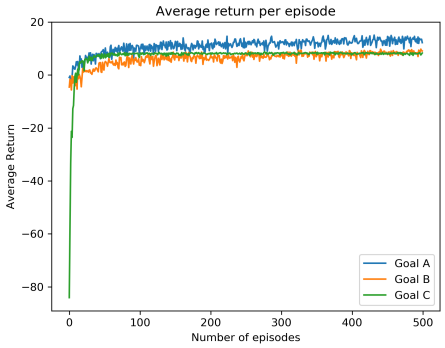
Figure 3: Average return per episode obtained by the agent in different world

From the fg:5, it clear that for initial few episodes agent takes longer time to reach the goal.

- nditemize

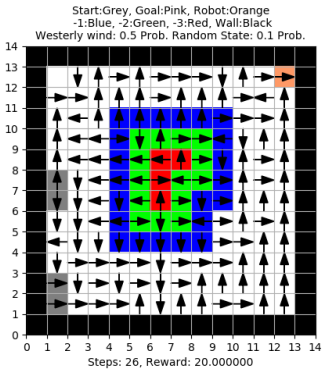


(a) Average steps

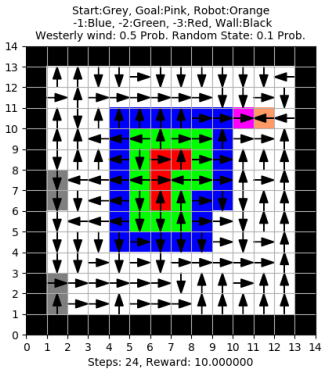


(b) Average return

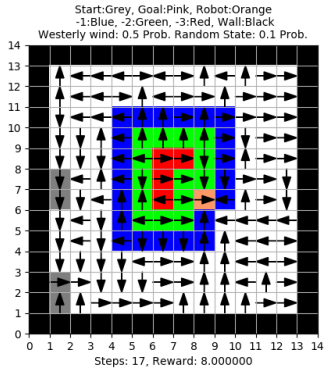
Figure 4: Combine plot of average return and average steps for different world



(a) Optimal Policy - A



(b) Optimal Policy - B



(c) Optimal Policy - C

Figure 5: Optimal Policy obtained after learning