Homework - 1 : DYNAMIC PROGRAMMING

KRITH SANVITH MANTHRIPRAGADA
KM59

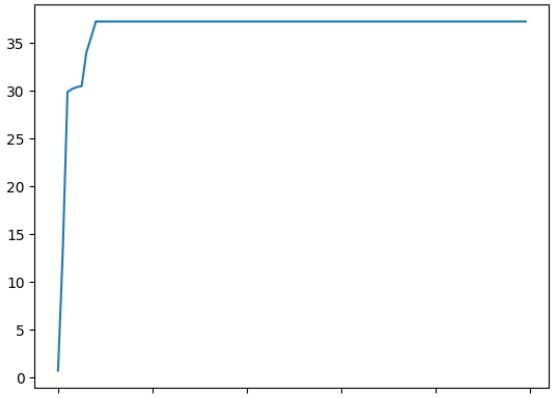
I. Introduction

THIS document contains results from four training agents which are Reinforcement Algorithms. Section two consists results from the gridworld environment and the next section consists results from the pendulum environment.

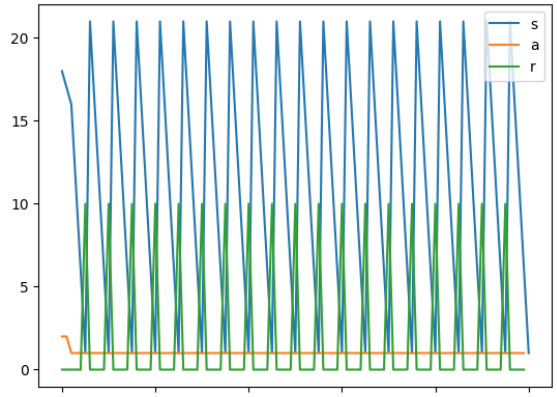
II. RESULTS FOR GRIDWORLD

A. Policy Iteration

[1]



[2]



[3]

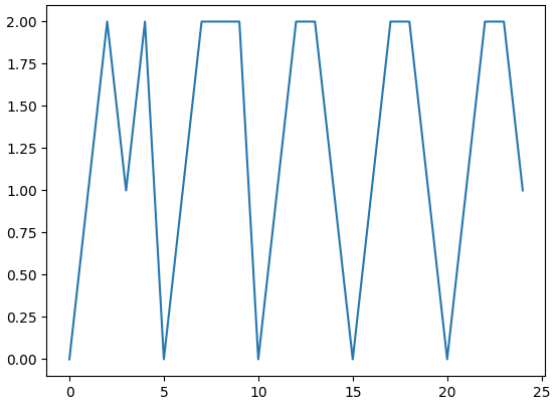


Fig. 1 (a) Mean value of the State Values function(b) Trajectory (c) Optimal Policy

The main thing to notice in the algorithm is the initialization, the first action for each state is being taken randomly with equal probability of all the actions. Here, I had to restrict the number of policy iteraions and value iterations because of the insufficient computational power. As expected the value of the state value function approached an optimal value after a certain number of iterations. It is evident that after attaining the optimal value function the trajectory is constant from there on so.

B. Value Iteration

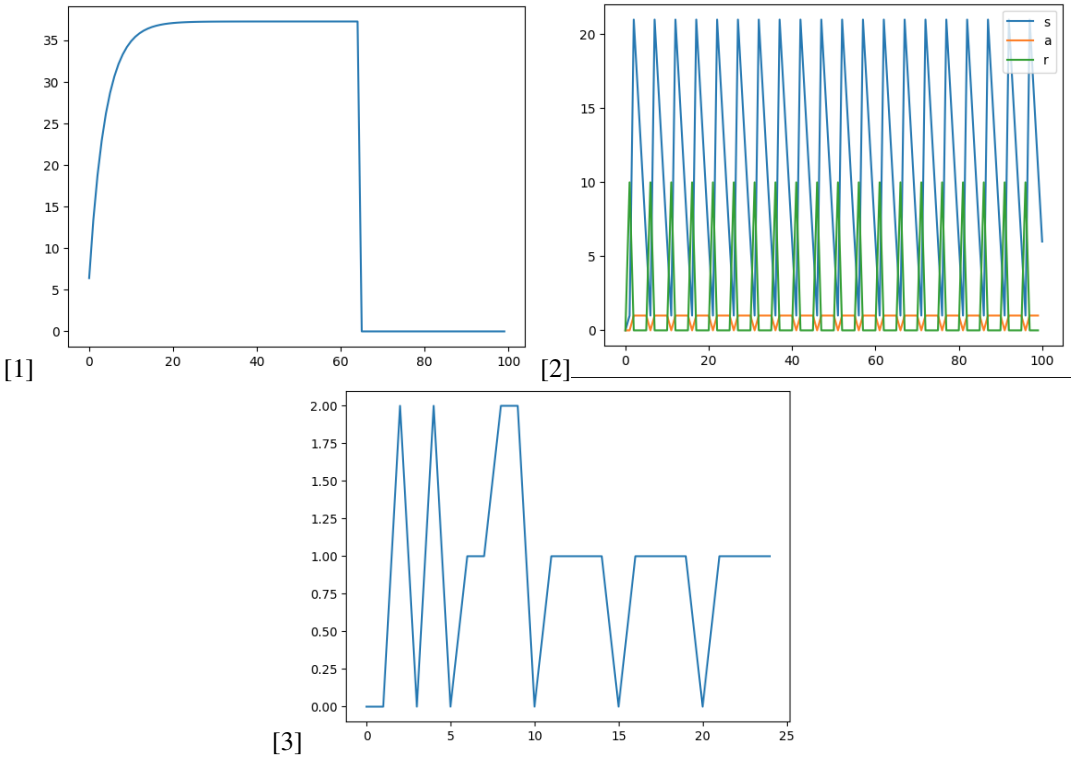


Fig. 2 (a) Mean value of the State-Value Function (b) Trajectory (c) Optimal Policy

In the value iteration algorithm also the initialization of the action space is in a similar manner, the probability of choosing the initial action for particular state is random with equal probability of every action. Coming to the plots, one might observe that the mean state value plot goes to zero, this is not due to the algorithms fault but we have given the initial values of the State value function as zero and after a certain number of iterations the loop breaks because of which the state value function doesn't go through the latter iterations, which inturn keeps the values functions value as zero. As we know we reach an optimal value function the trajectory and the policy remain in a similar sense.

C. SARSA

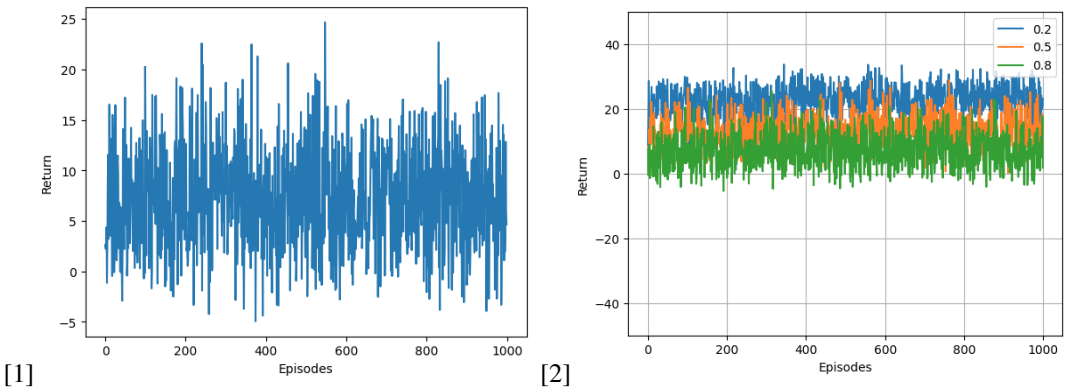


Fig. 3 (a) Return (b) Learning curve with varying epsilon

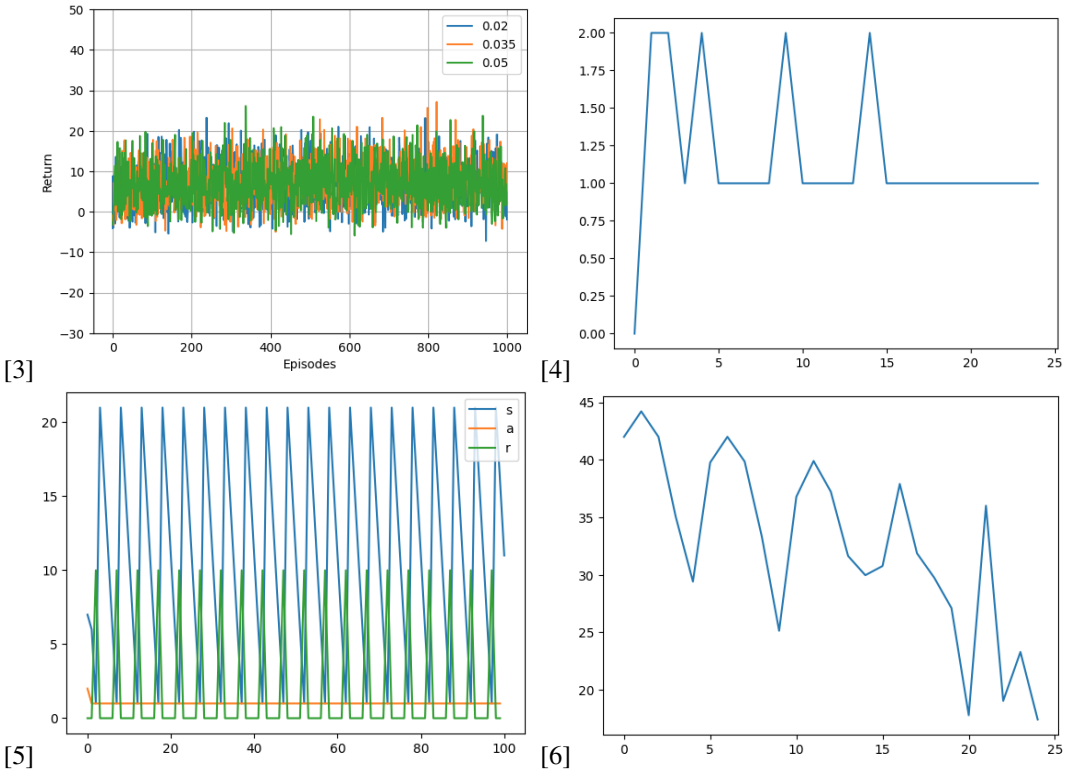


Fig. 4 (a) Learning curve with varying alpha (b) Optimal Policy (c) Trajectory (d) State-Value function(TD(0))

The Algorithm is pretty straight forward with action being taken using an epsilon-greedy policy. The first plot is the return values for different episodes we can see the return value is almost similar because of the policy being used. Coming to the learning curves with varying epsilon and alpha, we can see a significant change in the values of the return with varying epsilon values compared to alpha values. The Trajectory remains constant after reaching an optimal policy. To calculate the state value from TD(0) the action space is the arg max of the Action value value we obtain from SARSA

D. Q-Learning

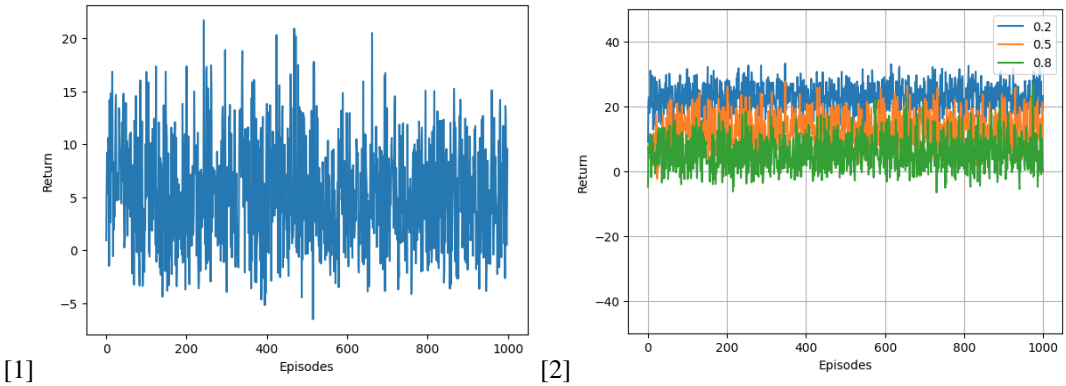
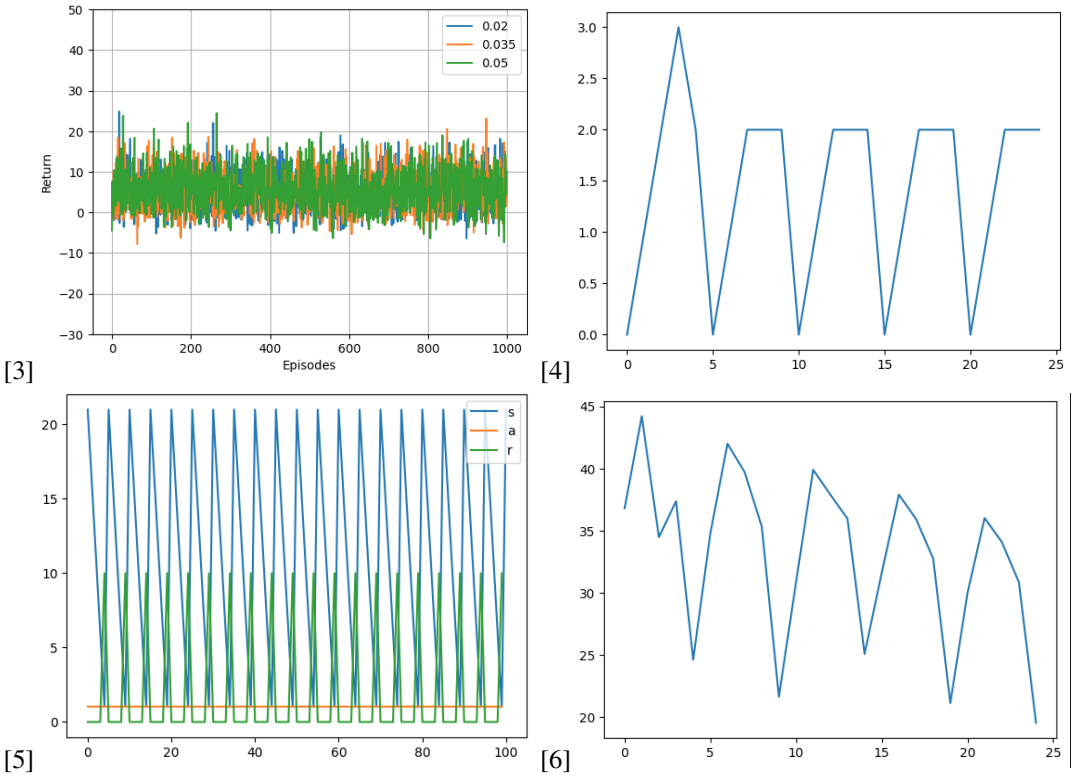


Fig. 5 (a) Return (b) Learning curve with varying epsilon



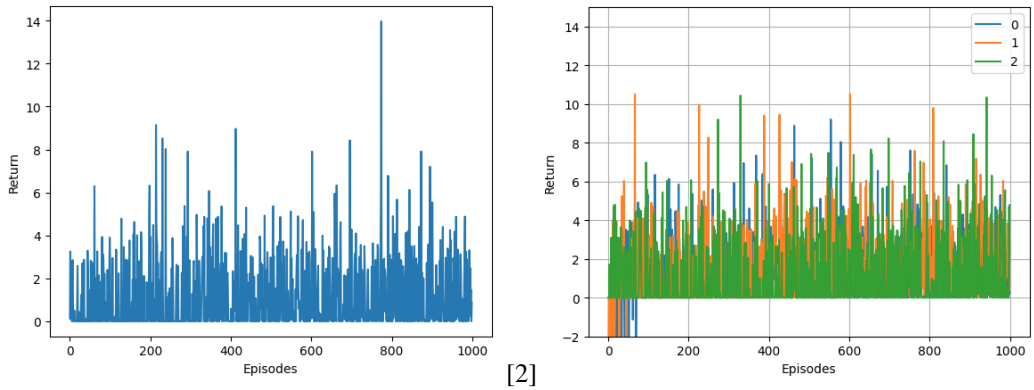
[3] [4] [5] [6]

Fig. 6 (a) Learning curve with varying alpha (b) Optimal Policy (c) Trajectory (d) State-Value function(TD(0))

Similar to SARSA, Q-Learning implementation is same except we take the action which gives max reward for the next state not using epsilon-greedy. Coming to the plots we observe similar variation of the return values and in the Q-learning case also the variation of return is higher compared to varying epsilon case than varying alpha case. Similar to SARSA after reaching an optimal policy the trajectory remains constant

III. RESULTS FOR PENDULUM

A. SARSA



[1] [2]

Fig. 7 (a) Return (b) Learning curve with varying epsilon

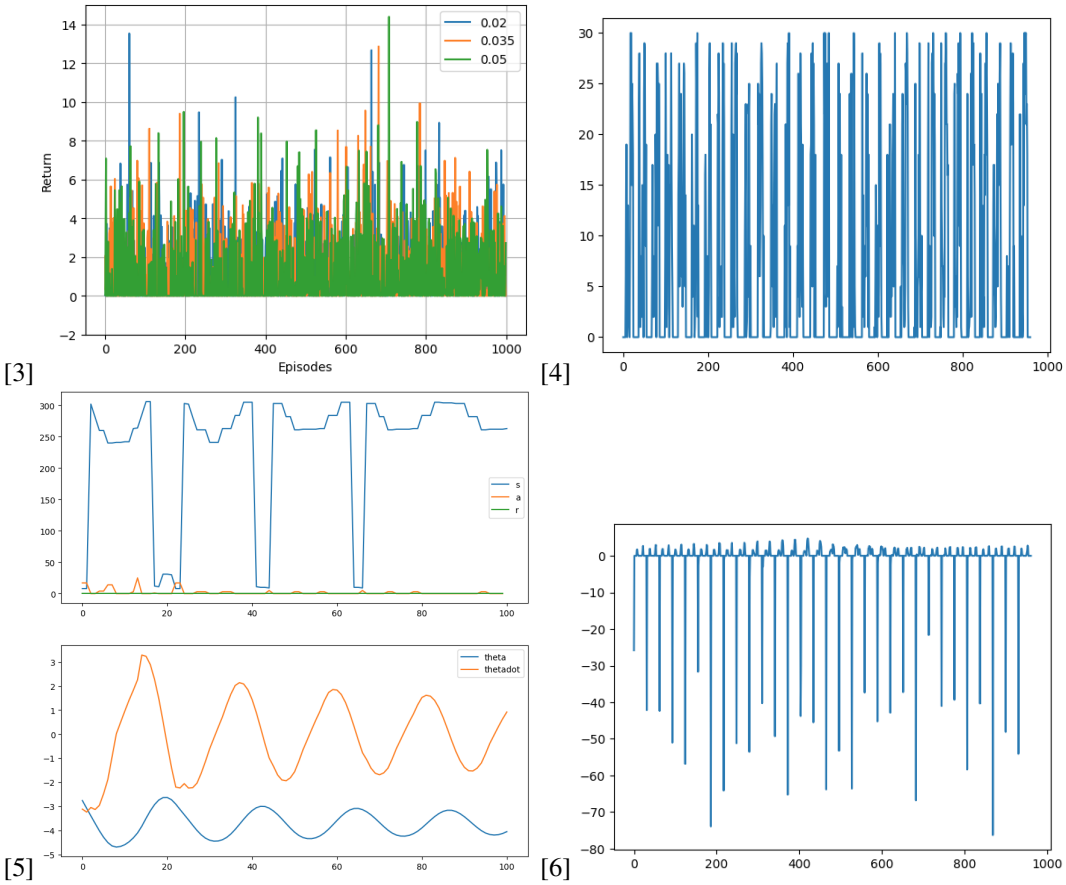
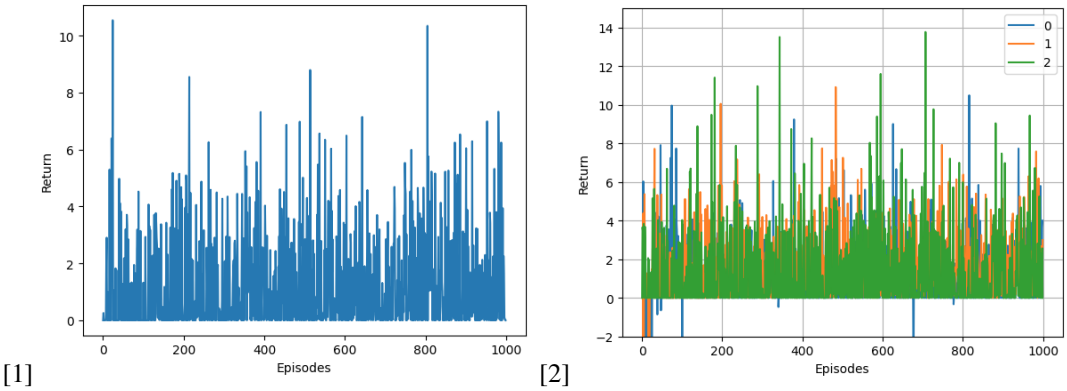


Fig. 8 (a) Learning curve with varying alpha (b) Optimal Policy (c) Trajectory (d) State-Value function(TD(0))

The values of the total return follow a similar fashion as in the gridworld case but here with varying epsilon and alpha we donot see a significant change in the return values. Trajectory looks as expected. Here due to the insufficient I have plotted for only three different values of epsilon and alpha, this is same for the Q-learning case as well.

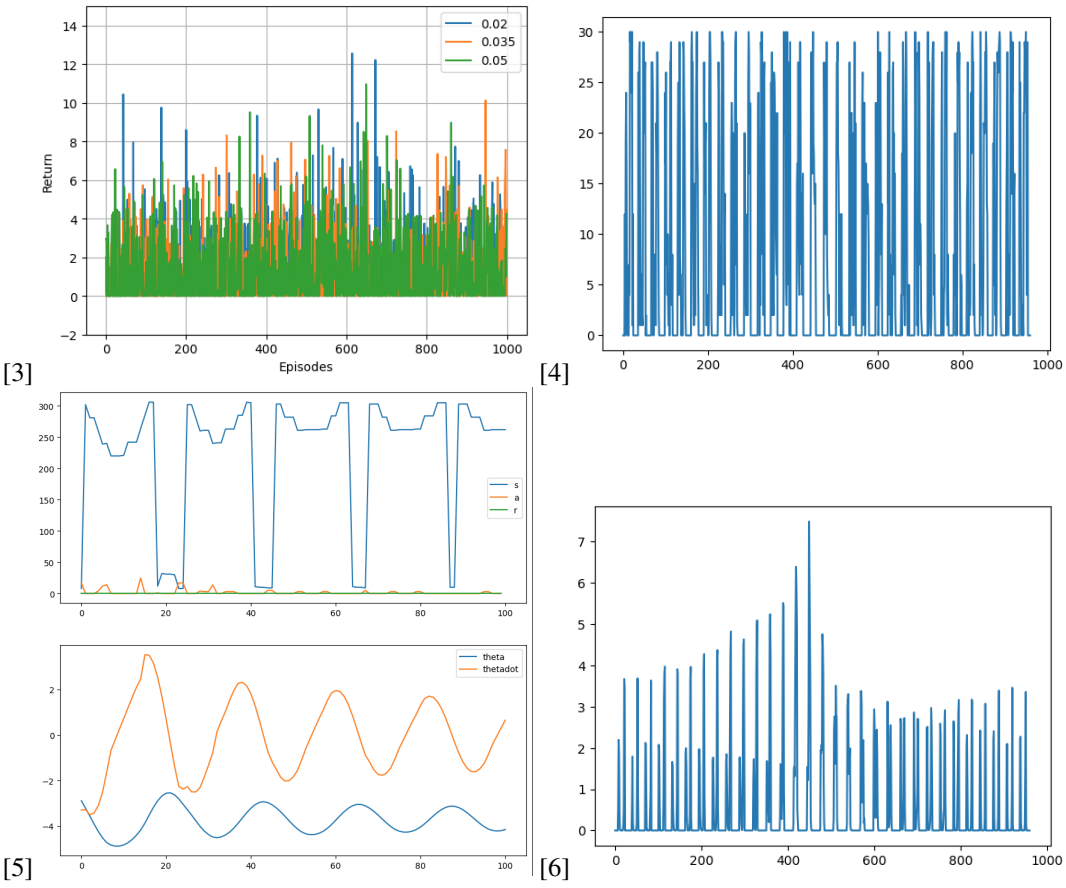
B. Q-Learning

The values of the total return follow a similar fashion as in the gridworld case but here with varying epsilon and alpha we donot see a significant change in the return values. Trajectory looks as expected.



[1] [2]

Fig. 9 (a) Return (b) Learning curve with varying epsilon



[3] [4]

[5] [6]

Fig. 10 (a) Learning curve with varying alpha (b) Optimal Policy (c) Trajectory (d) State-Value function(TD(0))