The Great Escape

Sri Datta Budaraju
budaraju@kth.se
No Partner

1 The Maze and The Minotaur

1.1 Formulating the MDP

The problem is formulated assuming general riles for both the Minotaur and the player. The constraints for the player are imposed in the rewards.

Let P_x, P_y, M_x, M_y denote the x and y positions of Player and Minotaur in the maze. And L, W be the length and width of the maze. $Dead$ and Win are terminal states which are reached after the player and Minotaur are in the same position or player is at the escape position B, while the Minotaur is not.

State Space S :

$\mathrm{s}_t = \{P_x, P_y, M_x, M_y\} \in S = [0, L] \times [0, W] \times [0, L] \times [0, W] \cup \{Dead\} \cup \{Win\}$

Actions :

$$\mathrm{A} = \{up, down, left, right, stay\}$$

Terminal Rewards :

$$\mathrm{r}_T(s = \text{Dead}) = 0$$
$$\mathrm{r}_T(s = \text{Win}) = 0$$

Non Terminal Rewards :

$$\mathrm{r}_t(s = \text{Dead}, a = \cdot) = 0$$
$$\mathrm{r}_t(s = \text{Win}, a = \cdot) = 0$$
$$\mathrm{r}_t(s = S_{\{P_x = M_x, P_y = M_y\}}, a = \cdot) = -1$$
$$\mathrm{r}_t(s = S_{\{P_x = B_x, P_y = B_y\}}, a = \cdot) = 1$$
$$\mathrm{r}_t(s = beside\_wall, a = towards\_wall) = -\infty$$

Time-horizon and objective:

$$\text{Finite horizon } N, \mathbb{E}\left\{\sum_{t=0}^{T-1} r_t\left(s_t, a_t\right) + r_T\left(s_T\right)\right\}$$

Transitions :

$$\mathrm{P}_t(s' = \text{Dead} | s = \text{Dead}, a = \cdot) = 1$$
$$\mathrm{P}_t(s' = \text{Win} | s = \text{Win}, a = \cdot) = 1$$
$$\mathrm{P}_t(s' = \text{Dead} | S_{\{P_x = M_x, P_y = M_y\}}, a = \cdot) = 1$$
$$\mathrm{P}_t(s' = \text{Win} | S_{\{P_x = B_x, P_y = B_y\}}, a = \cdot) = 1$$
$$\mathrm{P}_t(s' = S'_N | s = S_N, a = a_N) = \frac{1}{N}$$

Where S_N is a state with N number of neighbours and a_N is one of the valid actions that does not lead the Player or Minotaur outside the maze. That is at state S_N we have N possible states to move to and thus $1/N$ probability to go to each of the neighbour. Note that the $-\infty$ reward for Player makes sure that it can not enter the wall.

1.2 Derive the optimal policy with finite time horizon

Figure 1.2 illustrates the game played by an optimal policy for time horizon 20. The orange arrows with the number shows the action taken by the Player at time t and the purple ones shows that of the Minotaur. We can see that the player wins after 15 moves and the games reaches the Win state.

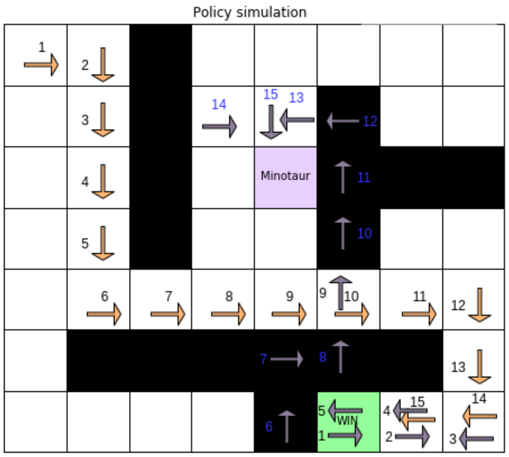

Figure 1: Game Illustration, Policy using Dynamic programming and Time Horizon 20

The plot on the left in figure 2 shows the trend in the maximal probability of escaping the make w.r.t time horizon T when the Minotaur cannot stay still. We can notice that the Player wins with probability 1 when the time horizon is 15 i.e the distance between start state to the goal state B. We can also notice that this would mean just taking the shortest route would always be safe. And this is purely because the placement of the agents is in such a way that they can not end up at the same position when Player takes the shortest path. We can see a less probability of escaping at T=15 when the Minotaur starts at (6,6) instead of (6,5).

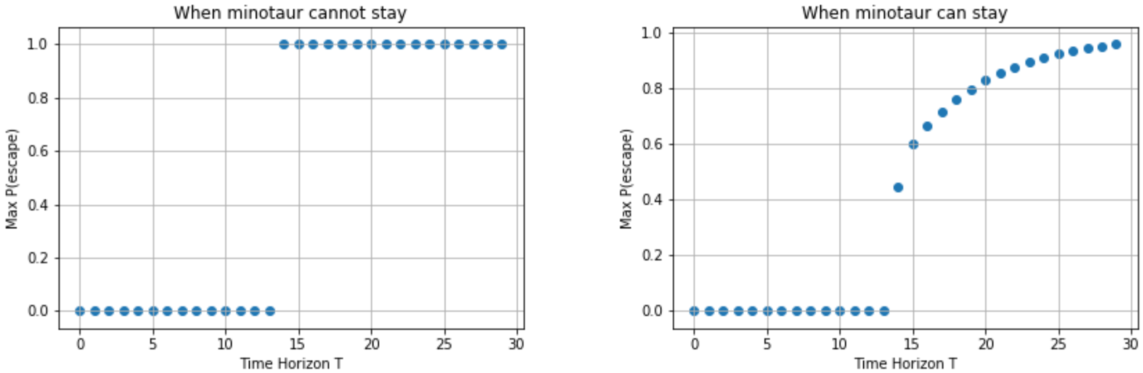

Figure 2: Trends in maximum probability of escaping the maze

However, allowing the Minotaur to stay at a place makes it much harder for the Player to safely escape the maze as illustrated in the plot to the right in the figure 2. Thus more likely to win with a longer horizon. For example, if both agents are face to face, the Player should move away and possibly towards the goal to be safe, if the Minotaur can stay. But if it can not stay the possibilities of its movement reduces to 4 and the best move is to move towards the Minotaur since it can not stay there. This would save a lot of steps making it easier to each the goal within the horizon.

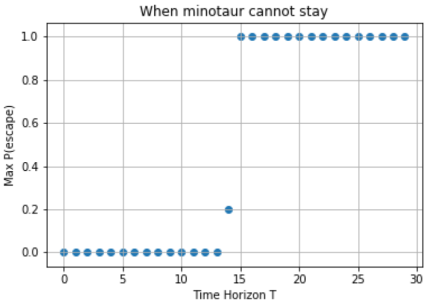

Figure 3: Maximum probability of escaping the maze vs Time Horizon T when Minotaur starting at (6,6) instead of (6,5)

1.3 Derive the optimal policy with infinite time horizon

To drive the optimal policy minimizing the expected time to escape the maze, we just have to change the model to discounted infinite time horizon.

Random Time Horizon: Time horizon T geometrically distributed with mean 30.

$$\mathbb{E}[T] = 1/(1-\lambda) = 30$$
$$\lambda = 29/30$$

Objective:

$$\text{Maximize } \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T} \lambda^{t-1} r\left(s_t^\pi, a_t^\pi\right)\right]$$

Simulating 10,000 games, the probability of escaping the maze is discovered to be 1. Hence with the given life distribution it is guaranteed to escape the maze alive.

2 Bank Robbing (Reloaded)

Similar to the maze problem, Let R_x, R_y, P_x, P_y denote the x and y positions of robber and police in the town. And L, W be the length and width of the town. $Caught$ and $Looting$ are states where the robber and police are in the same position or robber is at the bank B, while the police is not. And $start$ is the starting state.

State Space S :

$$s_t = \{R_x, R_y, P_x, P_y\} \in S = [0, L] \times [0, W] \times [0, L] \times [0, W]$$

No explicit state for $Caught$ or $Looting$

Actions :

$$\text{A} = \{up, down, left, right, stay\}$$

Rewards :

$$r_t(s = S_{\{R_x=P_x, R_y=P_y\}}, a = \cdot) = -10$$
$$r_t(s = S_{\{R_x=B_x, R_y=B_y\}}, a = \cdot) = 1$$

Time-horizon and objective:

$$Max\pi \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T} \lambda^{t-1} r\left(s_t^\pi, a_t^\pi\right) | s_1^\pi = start\right]$$

with $\lambda = 0.8$

Transitions :

$$\mathrm{P}_t(s' = S'_N | s = S_N, a = a_N) = \frac{1}{N}$$

Where S_N is a state with N number of neighbours and a_N is one of the valid actions that does not lead the police or the robber outside the town just as in the maze.

2.1 Q-learning

The figure 4 shows the improvement in the value function for different interesting states such as the start state and the state where robber and police are swapped etc. The Q value was updated for 10^7 steps where the value function converges.

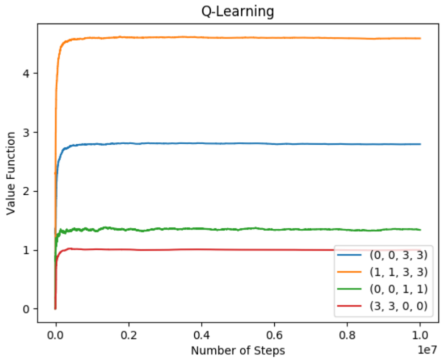

Figure 4: Q-learning

2.2 SARSA

The figure 5 and figure 6 shows the learning of the value function using SARSA with different ϵ for different interesting states same as that of Q-learning. The Q value was updated for 10^7 steps where the value function "tends" converges for certain values of ϵ, for certain states. Observing the plots for the start state, we can see that for ϵ 0.1 and 0.05 the learning is better but not better than that of Q-learning.

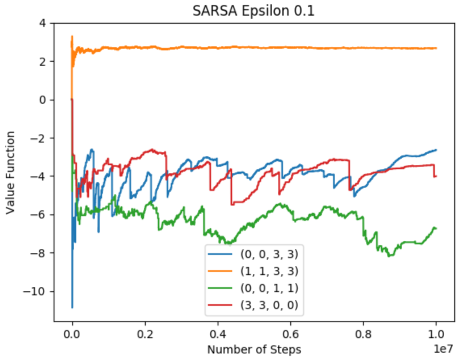

Figure 5: SARSA

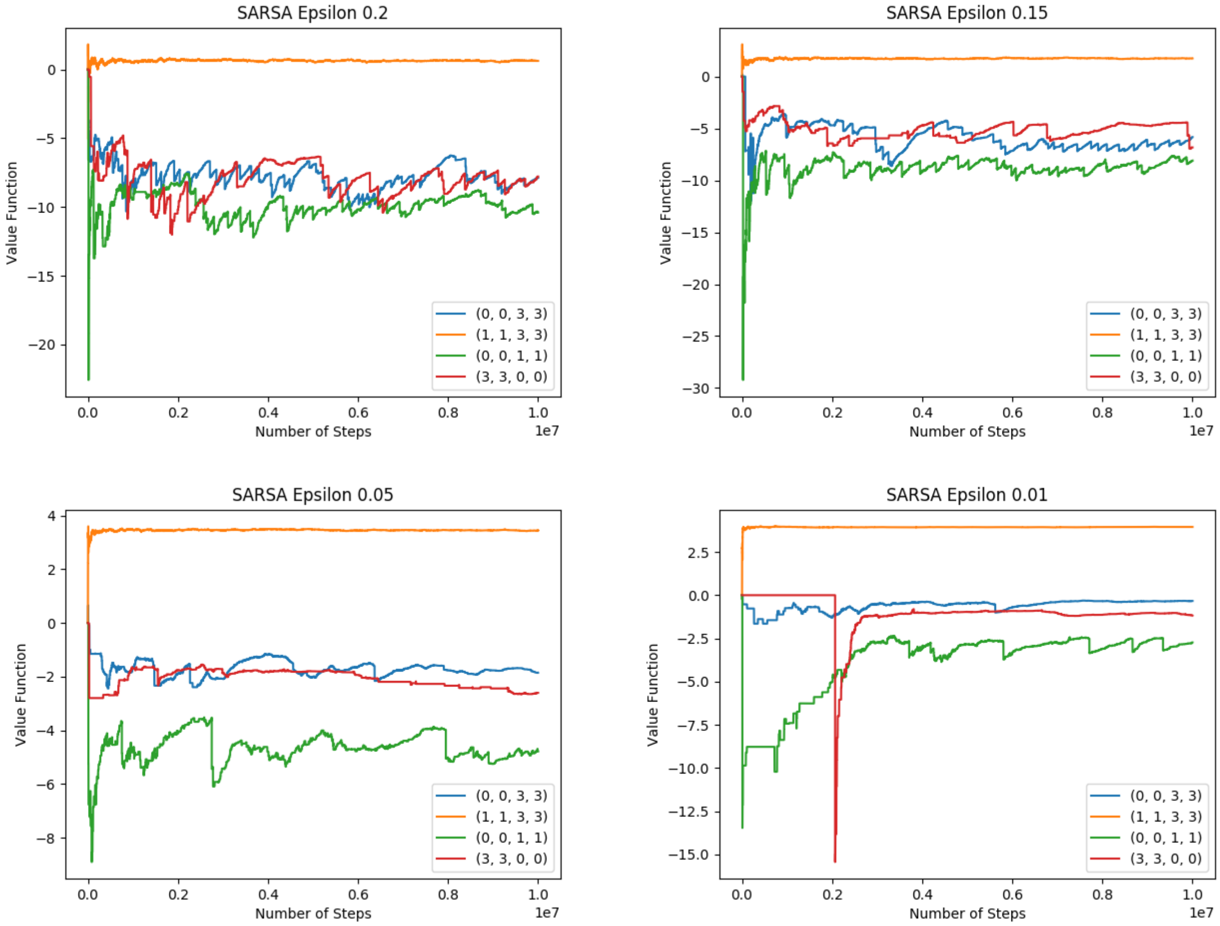

Figure 6: Additional Experiments with different ϵ