EQ2305 Lab1 report

Siyi Qian
qsiyi@kth.se
200012298709

Chenting Zhang
chzha@kth.se
200205146202

December 1, 2023

1 Problem 1

1.1 a

The state space consists of the player and Minotaur coordinates. The player can go anywhere besides obstacles, while the Minotaur can go to any grid in the maze. The state space is $S = \big\{(i,j) \times (k,l)$ such that (i,j) is not an obstacle$\big\}$. There are 8 horizontal and 7 vertical grids in the maze, including 16 obstacles. And there are two terminal states *lost* and *won* as the exit. Thus the total state number is $(8 \times 7 - 16) \times 7 \times 8 + 2 = 2242$ states

The action space is $A = \big\{$up, down, left, right, stay$\big\}$.

The rewards are:

- If at state s, taking action a, leads to a wall or an obstacle or the Minotaur, then $r(s,a) = -\infty$.
- If at state s, taking action a, leads to the exit, then $r(s,a) = 0$.
- If at state s, taking action a, leads to a position in the maze that is not the exit nor a wall nor an obstacle, then $r(s,a) = -1$.

The rewards here are independent of time.

The transition probabilities:

- If the player and the Minotaur are in the same position $s = (i, j, k = i, l = j)$, then the player is dead and cannot move to another position s'. The probability is $P(s'|s,a) = 0$.
- If the player arrives at the exit (ϕ_x, ϕ_y) without being caught, the player will stay at the exit. The probability is $P(s' = (i = \phi_x, j = \phi_y, k', l')|s = (i = \phi_x, j = \phi_y, k, l), a) = 1$.
- If the player is at state $s = (i, j, k, l)$, taking action a leads to a wall or an obstacle, and the player remains in this position. The probability is $P(s' = (i, j, k', l')|s = (i, j, k, l), a) = 1$.
- Define N_s as the number of possible moves of the Minotaur, which will be 4, 3, or 2 depending on the position of the Minotaur. The move is not dependent on the player's position or action. The transition probability is $P(s' = (i', j', k', l')|s = (i, j, k, l), a) = \frac{1}{N_S}$.

1.2 b

Given the situation where the player and the Minotaur are required to move alternatively and not in the same round, a signal *round* should be added to denote the round, in which 1 denotes the player's round and 0 denotes the Minotaur's round. The action space is the same as above.

The reward should be modified as follows:

- If at state s where round = 1, taking action a, leads to a wall or an obstacle or the Minotaur, then $r(s,a) = -\infty$.
- If at state s where (i,j) is one step near the final point and the Minotaur is not at the final point, plus round = 1, we take action a, leads to the exit, then $r(s,a) = 0$.
- If at state s besides the above possibilities, round = 1, taking action a, leads to a position in the maze that is not the exit nor a wall nor an obstacle nor Minotaur, then $r(s,a) = -1$.

The transition probabilities should be modified as follows:

- If the player and the Minotaur are in the same position $s = (i, j, k = i, l = j, round = 1)$, then the player is dead and cannot move to another position s'. The probability is $P(s'|s, a) = 0$.
- If the player arrives at the exit (ϕ_x, ϕ_y) without being caught, the player will stay at the exit no matter which round it is. The probability is $P(s' = (i = \phi_x, j = \phi_y, k', l') \cap round = 1 or 0 | s = (i = \phi_x, j = \phi_y, k, l), a) = 1$.
- If the player is at state $s = (i, j, k, l)$, taking action a leads to a wall or an obstacle, and the player remains in this position. The probability is $P(s' = (i, j, k', l')|s = (i, j, k, l, round = 1), a) = 1$.
- Define N_s as the number of possible moves of the Minotaur, which will be 4, 3, or 2 depending on the position of the Minotaur. The move is not dependent on the player's position or action. The transition probability is $P(s' = (i, j, k', l')|s = (i, j, k, l, round = 0), a) = \frac{1}{N_S}$.

Intuitively, we think that moving alternatively has more probability of being caught than moving simultaneously. Given the situation that the Minotaur is exactly adjacent to the player if moving simultaneously, the player has a chance of survival by avoiding jumping to the Minotaur. However, if moving alternatively, the play has an absolute 0.25 probability of being eaten. In this situation, simultaneously moving has a higher survival rate.

1.3 c

In this finite state situation when T=20, we implement dynamic programming to solve the problem. Two different scenarios are taken into consideration, the Minotaur can stay or not and we reflect this variable in the available actions for the Minotaur to take. For each action the player chooses, there are N_S possibilities and N_S states. Each state-action pair have a corresponding reward function. Here we assume the cost of meeting a Minotaur within its reach is the same as of meeting an obstacle. The action-making procedure of the player is determined by the average of each possible reward.

Figure 1 illustrated one possible path of the trajectory of the player as well as the destination of the minotaur. Figure 2 illustrated two possible paths of the player as well as the destination of the minotaur. It can be seen from the figure that there are possibilities the player could not arrive at the exit if the minotaur can stay.

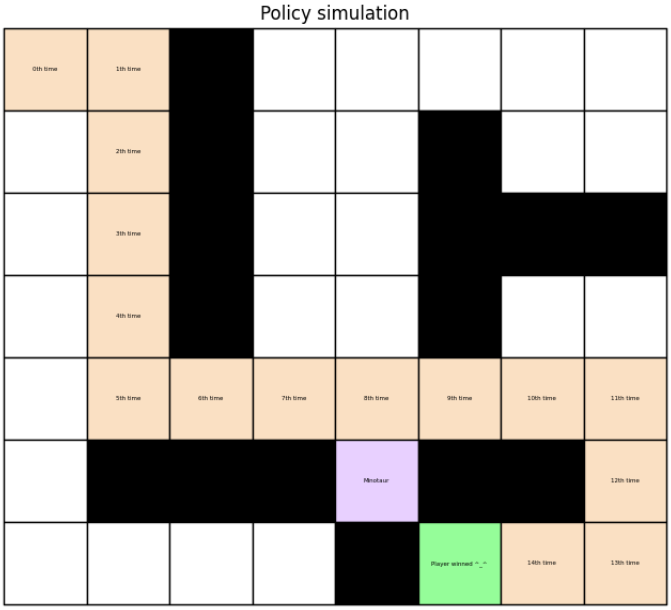

Figure 1: Minotaur must move in every time instance

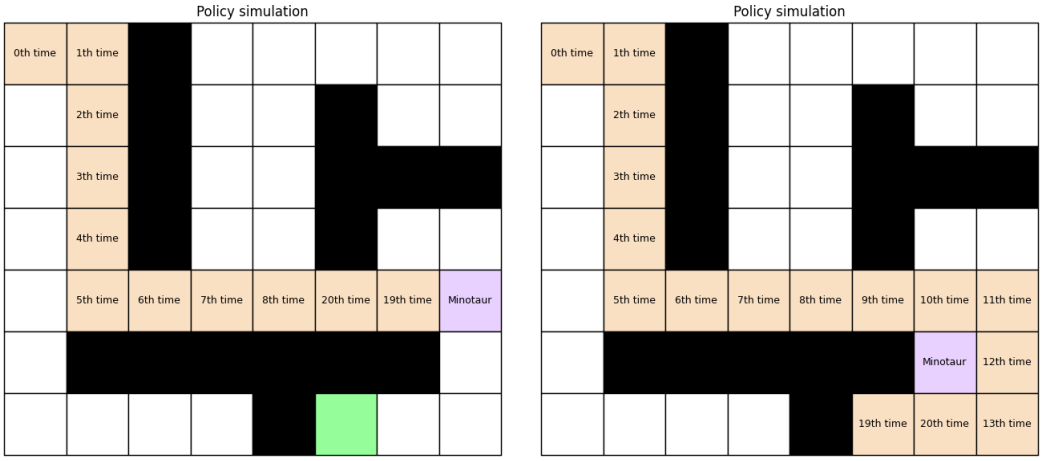

Figure 2: Minotaur can stay in each time instance

1.4 d

Figure 3 illustrates the success rate for $T = 1 \dots 30$, we run 10000 times for simulation, and calculate the success rate that the player could arrive at the exit.

We can observe from the figure that in both situations, the player has no chance to win and the success rate is 0 when $T < 14$. However, when $T > 14$, there always exists one optimal policy that the player can reach the destination if the Minotaur move simultaneously because if the Minotaur is adjacent to the player, the player knows that the Minotaur will move thus it can run into the square where the Minotaur used to be. However, if the Minotaur can choose to stay, there will be a cost of walking towards the location of the Minotaur thus the player cannot make a decision freely. There exists the possibility that the Minotaur chooses to stay and the player cannot arrive at the exit in a limited amount of time.

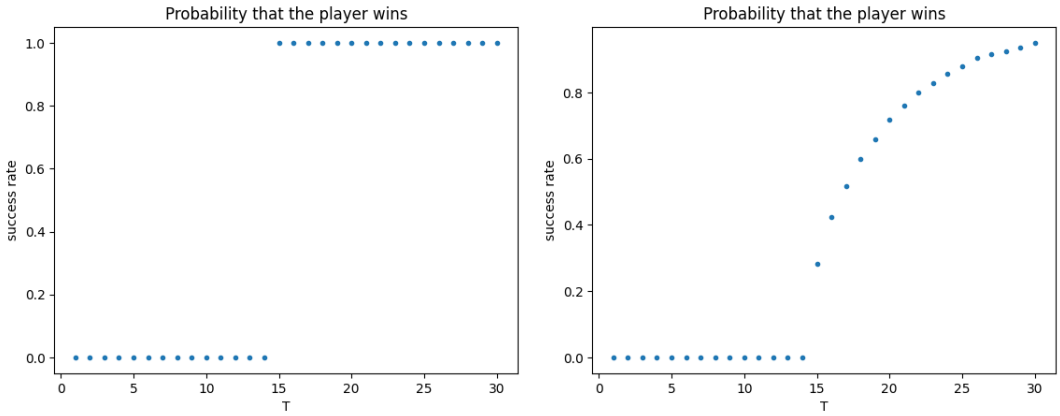

Figure 3: Minotaur can stay in each time instance

1.5 e

In the problem statement, the life expectancy due to poisoning is described as geometrically distributed with a mean of 30. This suggests that there is a probability p such that the expected number of periods until "failure" (in this case, death from the poison) occurs is 30. The expected survival time could be written as:

$$E(T) = \frac{1}{p}. \tag{1}$$

This problem can modelled as an infinite-time horizon MDP. The Objective is to find a policy π over all possible policies to maximise equation(2). In here, the discount factor $\lambda = 1 - p$. Thus, we set

$\lambda = \frac{29}{30}$ and $\epsilon = 0.0001$ for our policy generation. Figure 4 illustrates one random result of our policy simulation under this scenario. In this run, the player is poisoned to death for the 7th time.

$$E\Big[\sum_{t=0}^{\infty} \gamma^t r\big(s_t, \pi(s_t)\big)\Big]. \tag{2}$$

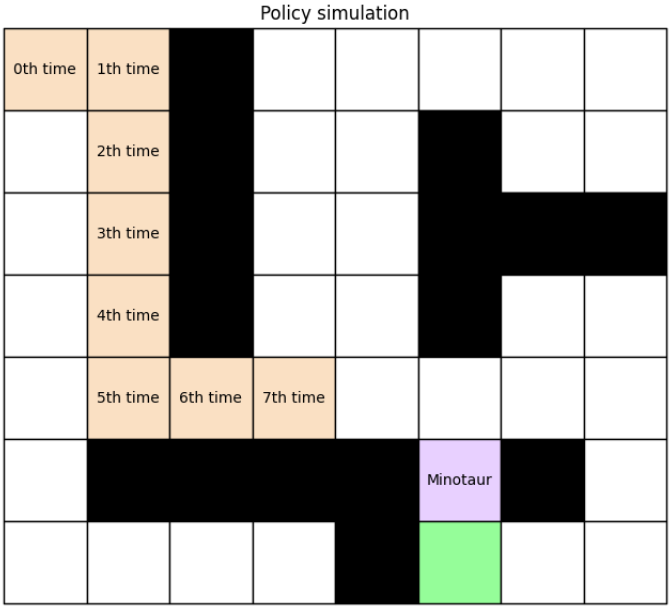

Figure 4: One run of value iteration

1.6 f

We ran the simulation 10000 times and evaluated the rate at which the player could make it alive to the exit. Since we chose the method Value Iteration, the policy generated does not contain the dimension time. If the minotaur is forced to move, the success rate is 61.77%. If the minotaur could stay, the success rate is 55.11%. The intuitive reason for this is if the minotaur could stay, the expected time cost for the player to make it to the exit is longer than when the minotaur must move, thus the survival rate is correspondingly lower due to the poison

1.7 g

1

On-policy learning is a method where the agent learns the value of the policy it is currently following. That means the agent has to explore the environment and try different actions to improve its policy. Target Policy == Behavior Policy. Off-policy learning is a method where the agent learns the value of a different policy than the one it is following. That means the agent can use data from other sources or policies to improve its policy. Target Policy != Behavior Policy.

2

For Q-Learning, the convergence conditions are stated as follows:

$$\sum_{t=1}^{\infty} \alpha_t(s,a) > \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2(s,a) < \infty$$

Under this condition, $Q^{(t)}$ will converge to Q^* as t goes to ∞.
For SARSA, the convergence conditions are stated as follows:

$$\sum_{t=1}^{\infty} \alpha_t(s,a) > \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2(s,a) < \infty$$

Under this condition, $Q^{(t)}$ will converge to Q_π as t goes to ∞, where π is the chosen bahavior policy.

1.8 h

Firstly, this scenario could be modeled as infinite MDP. Since the player must get the key to exit, compared to the MDP we modeled previously, we set two mazes. In the first maze, the starting point is at A and the arrival point is at C. In the second maze, the starting point is at C and the arrival point is at B. The player's life is geometrically distributed with mean 50 which means the discount factor $\lambda = \frac{49}{50}$. We fetched the time cost t in the first round to get the key as well as the location of the Minotaur (k, l) to feed into the input parameter in the next round to exit.

Additionally, in this scenario, the Minotaur no longer moves randomly. Instead, it has a higher probability of moving towards the player. Here we make our own definition of how Minotaur will move toward the player as follows:

- Suppose the player is at position (1,1) and Minotaur is at (4,5). The player is now at the left-up corner of Minotaur, so Minotaur will have a possibility of $\frac{0.35}{2} + \frac{0.65}{4} = 0.3375$ to go left, 0.3375 to go up, 0.1625 to go right, and 0.1625 to go down.
- Suppose the player is at position (1,1) and Minotaur is at (1,3). The player is now at just the left side of Minotaur. So Minotaur will have a possibility of $0.35 + \frac{0.65}{4} = 0.5125$ to go left and 0.1625 to go to the remaining three directions.

The state space, action space, and the reward functions are the same as we described in part a. The transition probabilities:

- If the player and the Minotaur are in the same position $s = (i, j, k = i, l = j)$, then the player is dead and cannot move to another position s'. The probability is $P(s'|s, a) = 0$.
- If the player arrives at the exit (ϕ_x, ϕ_y) without being caught, the player will stay at the exit. The probability is $P(s' = (i = \phi_x, j = \phi_y, k', l')|s = (i = \phi_x, j = \phi_y, k, l), a) = 1$.
- If the player is at state $s = (i, j, k, l)$, taking action a leads to a wall or an obstacle, and the player remains in this position. The probability is $P(s' = (i, j, k', l')|s = (i, j, k, l), a) = 1$.
- Define N_s as the number of possible moves of the Minotaur, which will be 4, 3, or 2 depending on the position of the Minotaur. Define N_t as the number of possible moves of the Minotaur moving toward the player, which will be 2 or 1 depending on the relative position of both. The transition probability is $P(s' = (i', j', k', l')|s = (i, j, k, l), a) = \frac{0.35}{N_t} + \frac{0.65}{N_s}$ if (k', l') is closer to (i, j) than (k, l) and $P(s' = (i', j', k', l')|s = (i, j, k, l), a) = \frac{0.65}{N_s}$ vice versa.

One example of these two rounds is illustrated in figure 5

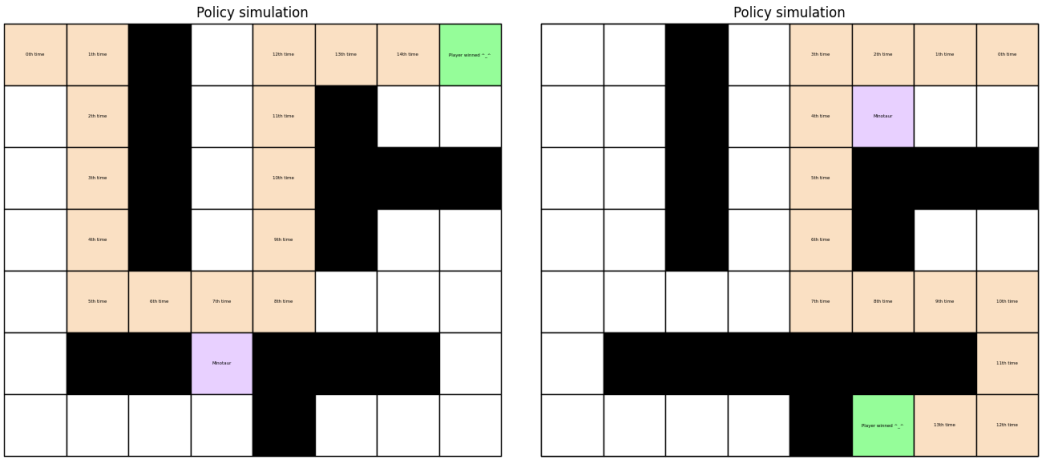

Figure 5: The paths of the two rounds for the player

By adding the total time in these two rounds, we know that it took 29 times movement for the player to win. We also calculated the success rate by doing the simulaton 10000 times. It turns out that

although the player is slightly poisoned, the success rate under the circumstance that the minotaur must move is 58.40% due to the minotaur has more probability to move towards the player as well as the necessity of getting the key for success.