

CSC384H1: Assignment 2, Written Part

Junming Zhang

Due: June 18th, 2019 before 10 p.m.

Question 2: Minimax

1. Note that Pacman will have suicidal tendencies when playing in situations where death is imminent. Why do you think this is the case? Briefly explain in **one or two sentences**. (2 points total)

Solution.

Due to the principle of Minimax algorithm, when the Pacman is moving, it is the **MAX** player and eating the pacman gains a higher score from eating a ghost during the scare time of the ghost. In this case, the Pacman (i.e., the **MAX** player) will choose the option for getting the higher mark, which is the reason why it looks like "suiciding".

Question 3: Alpha-Beta Pruning

1. You should notice a speed-up compared to your **MinimaxAgent**. Consider a game tree constructed for our Pacman game, where b is the branching factor and where depth is greater than d . Say a minimax agent (without alpha-beta pruning) has time to explore all game states up to and including those at level d . At level d , this agent will return estimated minimax values from the evaluation function.
 - (a) In the best case scenario, to what depth would alpha-beta be able to search in the same amount of time? (1 point total)

Solution.

In the best case scenario, the alpha-beta can search up to the depth $2d$. Since minimax method have to search $\mathcal{O}(b^d)$ nodes to find the goal value, however, the alpha-beta pruning only need searching $\mathcal{O}(b^{\frac{d}{2}})$ nodes to find the same goal in the best case, which means in the same time, alpha-beta pruning can reach the twice depth as what minimax without pruning can theoretically.

- (b) In the worst case scenario, to what depth would alpha-beta be able to search in the same amount of time? How might this compare with the minimax agent without alpha-beta pruning? (2 points total)

Solution.

In the worst case scenario, the alpha-beta can search up to the depth d , the same as what minimax without pruning can. Because in the worst case, no pruning can be made, that means, in the pseudo-code, the line

```

if  $\beta \leq \alpha$  then
  break
end if

```

will never be executed, that is, $\beta > \alpha$ for all the time for both beta-cut and alpha-cut. Therefore, alpha-beta pruning explores the same number of nodes as minimax without pruning. Minimax without pruning search $\mathcal{O}(b^d)$ nodes to find the goal value in the worst case as well, thus alpha-beta pruning also search $\mathcal{O}(b^d)$ nodes. Therefore, it reaches the depth d , the same as minimax without pruning, in the worst case.

2. **True or False:** Consider a game tree where the root node is a max agent, and we perform a minimax search to terminals. Applying alpha-beta pruning to the same game tree may alter the minimax value of the root node. (1 point total)

Solution.

False. Because alpha-beta pruning is used to cut the path not needed to search, that is, the value in that path is impossible to be selected as min/max value for the upper level.

Question 4: Expectimax

1. Consider a game tree where the root node is a max node, and the minimax value of the tree is v_M . Consider a similar tree where the root node is a max node, but each min node is replaced with a chance node, where the expectimax value of the game tree is v_E . For each of the following, decide whether the statement is **True** or **False** and briefly explain in **one or two sentences** your answer.

- (a) **True or False:** v_M is always less than or equal to v_E . Explain your answer. (2 points)

Solution.

True. Since in the expectimax algorithm, the MIN player is replaced by CHANCE player, it means the min node is possibly selected, not always selected. And the MAX player is still kept. Thus the max child is always selected. Therefore $v_E \geq v_M$.

- (b) **True or False:** If we apply the optimal **minimax** policy to the game tree with chance nodes, we are guaranteed to result in a payoff of at least v_M . Explain your answer. (2 points)

Solution.

True. In the expectimax algorithm, the max child is always selected, and the min child is possibly to be selected. To choose the minimum, CHANCE player will always choose the child with the minimum value, which results in v_M .

- (c) **True or False:** If we apply the optimal **minimax** policy to the game tree with chance nodes, we are guaranteed a payoff of at least v_E . Explain your answer. (2 points)

Solution.

False. Based on the brief analysis in part b), the guaranteed least payoff is v_M , which is less than or equal to v_E , if the CHANCE player always choose minimum child and MAX player still choose the maximum, the value finally selected is v_M .