ECSE – 526

Artificial Intelligence

Vraj Patel

(McGill ID: 261022581)

Abstract: In this report, I've tried to show my result and analysis for two algorithms namely, **Minimax** and **α-β pruning** for dynamic connect-3 on two different sizes of boards. Both algorithms are used for two player zero sum game. There are three different tasks which are described in the report. In the first task, the game was played on 5X4 board size. I've shown results of different depth of the search tree with respect to time and the results of α-β pruning with compared to minimax algorithms. In the task-2, I've mentioned the effect of simple heuristic function and it's effect on the performance of the algorithm and lastly, in the third task I've shown result for bigger board of 7X6 and it's performance with respect to both the algorithms

Introduction:

Minimax and **α-β pruning** are simple search algorithms. They are used for two player zero sum game. One player is trying to maximize its outcome value at the same time other player tries to minimize its outcome value. Here, in my program AI can play as white as well as black piece and I've designed my program such that white player always tries to maximize its outcome meanwhile, black player minimize it's outcome.

Minimax: Minimax is a simple algorithm, it tries to search over all the possible moves of place and according to that it'll try to maximize or minimize the outcomes of the player. I've designed a heuristic function which will give me the heuristic value on that node. Since, minimax will try to search through all the possible outcomes, there is a need to cut off to certain depth of the tree and try to calculate the value at that node. The performance of the algorithm largely depends on to what extent we go deeper into the tree. Deeper we go better would the

performance of the algorithm. On the contrary, it increases the computing power of the program.

α-β pruning: This algorithm is more efficient than minimax since, it reduces the computing power of our algorithm by pruning the unnecessary branches which doesn't lead winning a game. α-β pruning introduces two important parameters α and β. With the use of these parameters, we can reduce the time of running a program by preventing our algorithms to iterate to unnecessary outcomes of the game and make the performance more robust.

Heuristic function: Heuristic function is the integral part for both the algorithms since, we prevent our algorithm to go over all the search space, we need to use certain heuristic to calculate the value of the node and from that value we can make a decision about which path our AI should take which gives it better probability to win a game. Better the heuristic function, more is the chance of winning a game.

Design of the game: For this assignment, we're given two types of game board, 5X4 and 7X6. At the same time, I've designed my program as per given guidelines by the instructor. All the starting positions and notation for move is as per instructions.

Task-1:

A. Analysis of different depth of the search tree and time taken to make a move for minimax algorithm.

For this task I've taken different depths of the tree ranging from depth = 2 to depth = 4. Since, more the number of depths are more will be time taken by an agent to make a move.

Table 1.A is the time taken by an agent to make a move for different depths of the tree. Fig 1.A is the graph of average time V/S number of depths of the tree.

Time taken by Minimax algorithm at different depths

Depths	Average time taken(sec)
2	11.82
3	17.20
4	18.71
5	20.88
6	24.043

Table 1.A

Minimax (depth v/s time)

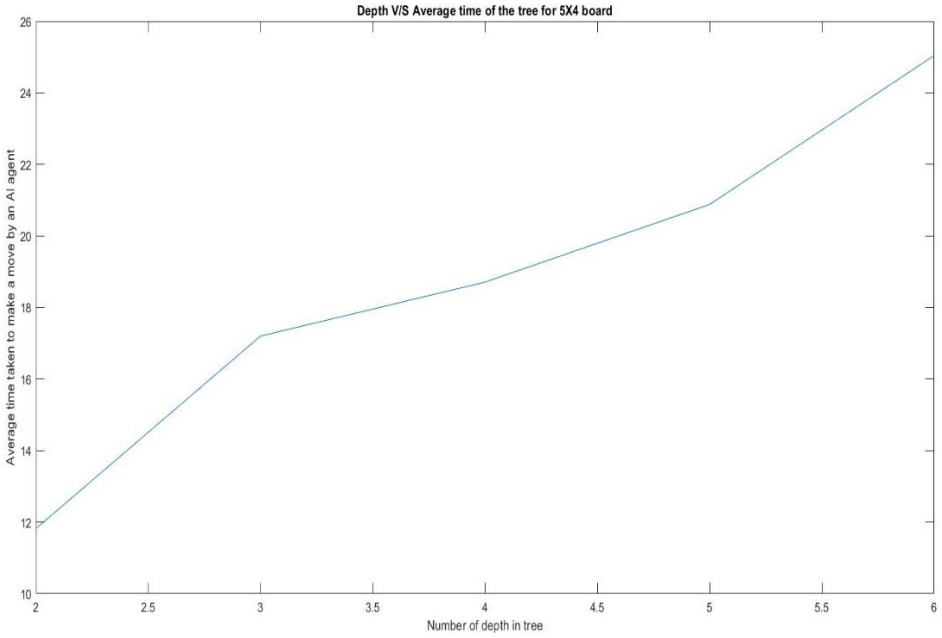

Fig 1.A

From the above table 1.A and Fig 1.A , we can say that more we go deeper in the tree more computation time would be required by the program.

B. Performance of α-β pruning with comparison with minimax search algorithm.

Knowing that, minimax algorithm is very slow because it takes too much time by going over all the possible outcomes, we improve that part by modifying minimax with α-β pruning with stop going over to those nodes which leads us to loss. Meaning, in the latter we check which path leads us for favorable win, we'll end up at those nodes where we have maximum probability of winning the game. By doing this we reduce our computing power to some extent and make agent's actions more robust.

Time taken by α-β pruning algorithm at different depths

Depths	Average time taken(sec)
2	10.05
3	11.61
4	13.63
5	18.88
6	20.65

Table 1.B

α-β pruning (depth v/s time)

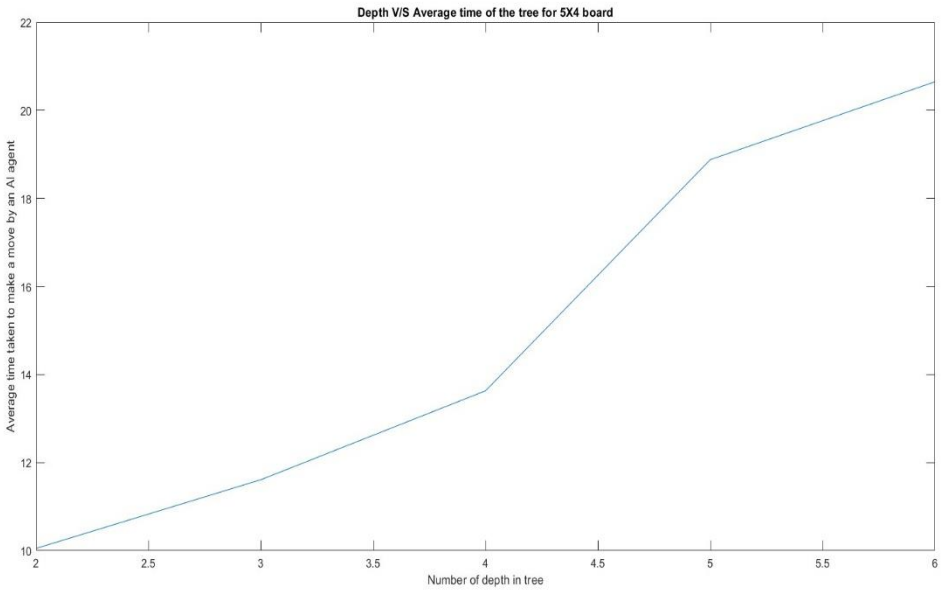

Fig 1.B

Comparison of minimax and α-β pruning

Depths	Minimax	α-β pruning
2	11.82	10.05
3	17.20	11.61
4	18.71	13.63
5	20.88	18.88
6	24.043	20.65

Table 1.C

Table 1.B shows the performance of α-β pruning algorithm on 5X4 board and Fig 1.B shows the depth v/s time taken to make a step by an agent.

Table 1.C shows comparison of minimax and α-β pruning for different depths. From the data itself, we can say that for each depth latter takes lesser time than minimax because of the pruning effect.

Task -2

C. Discussion of the Heuristics function

Definition of heuristic function:

It is shorter way to solve a problem when we don't know the end solution. In our way, we don't go deeper into end nodes of the all the possible outcomes of the given state, we simply stop to a certain depth and calculate the value of the heuristic function and return this to parent nodes up to root node to figure out the best possible route which gives the highest probability of winning a game.

Heuristic Function:

H(n) = (number of runs of 2 white pieces - number of runs of 2 black pieces)

For my algorithm I've used simple heuristic function which was given in the assignment description.
If two consecutive white pieces are in line then return a value of 1, if there are more than two consecutive white pieces then return higher value than 1(in my program I've returned 100). Same applies for black pieces.
I've subtracted white's value from black's value since white is maximizing player.

Importance:

Heuristic function plays pivotal role in playing style of an agent. Better is the heuristic more is the probability of winning a game. At the same time, heuristic's goal is to make decision making faster so it'll also make program faster and more efficient.

Task -3

Running the same algorithms on 7X6 board

D. Minimax and α-β pruning on 7X6 board

Comparison of Minimax and α-β pruning 7X6 Board

Depths	Minimax	α-β pruning
2	22.83	16.54
3	25.84	23.03
4	31.01	26.93
5	37.65	33.90
6	46.48	36.92

Table 3.D

Depth v/s average time for Minimax

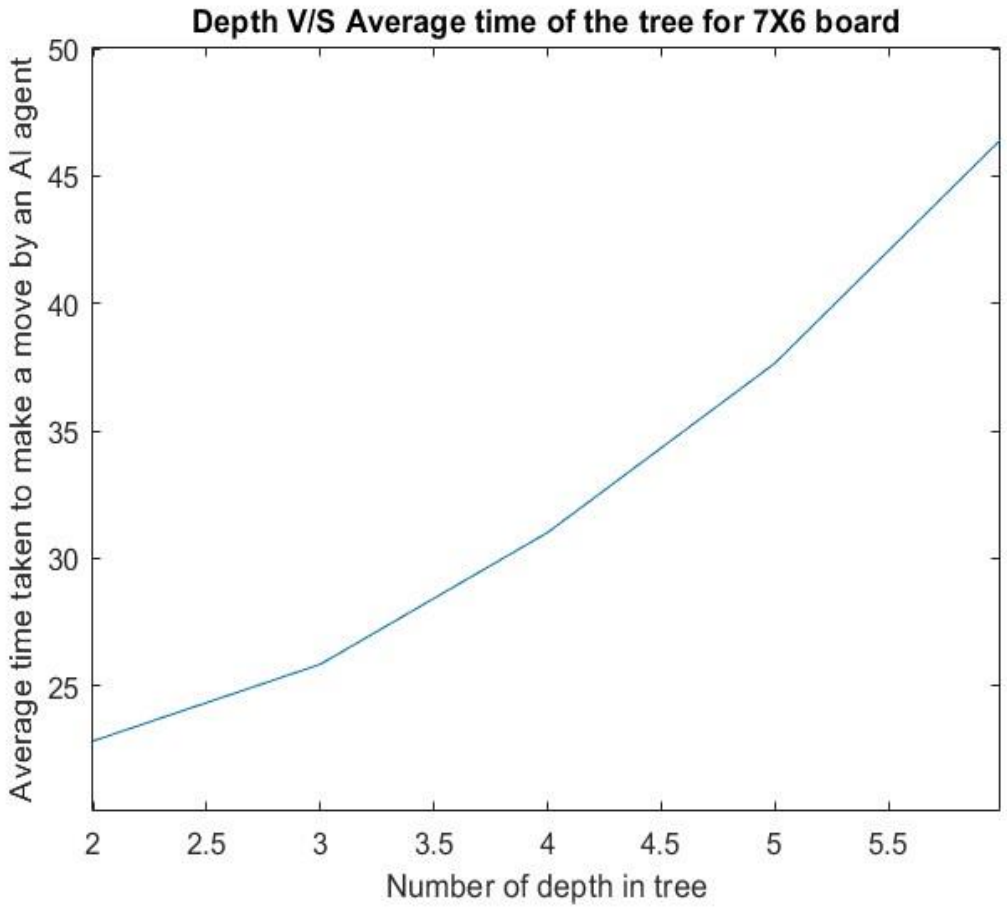

Fig 3.D(1)

Depth v/s average time for α-β pruning

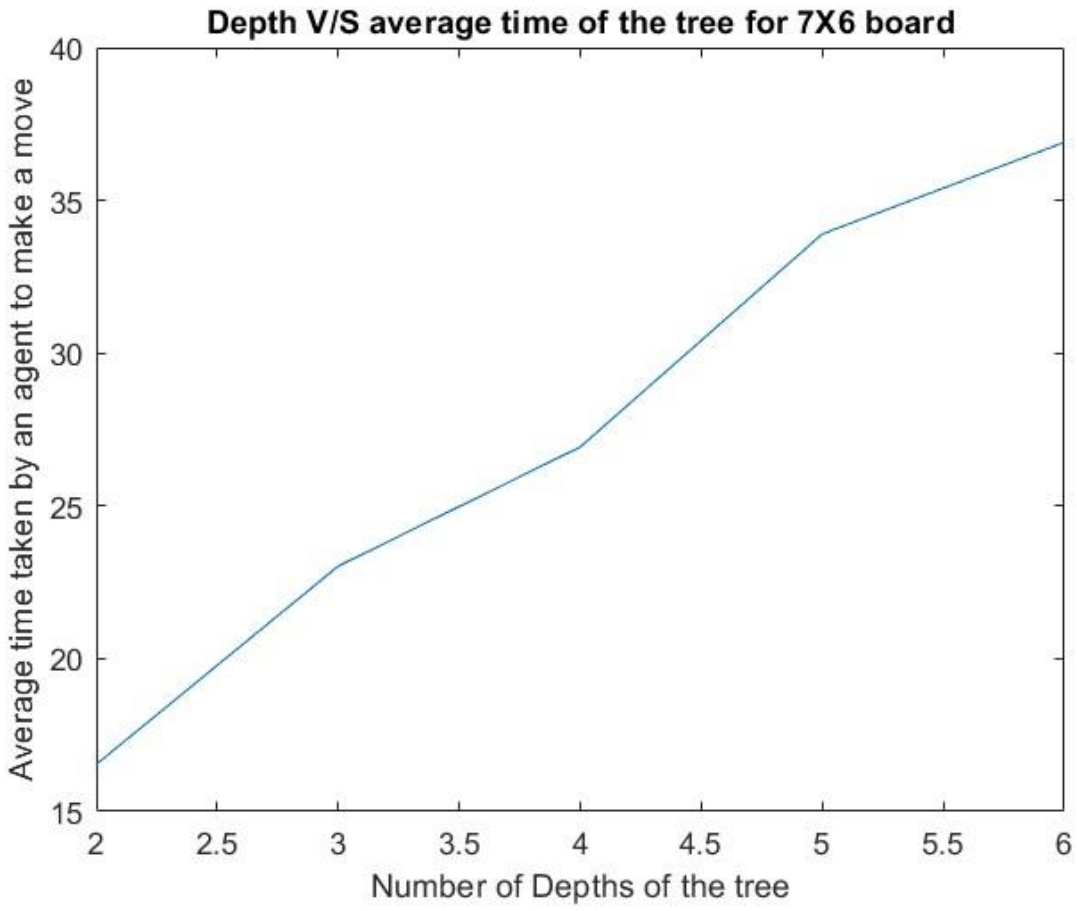

Fig 3.D(2)

Table 3.D, Fig 3.D(1) and Fig 3.D(2) all are the performance of the AI agent on different algorithms on 7X6 board.

Conclusion: In order to improve the performance of AI agent we should always use α-β pruning over minimax algorithm, because it's way faster than minimax. Also, heuristics function play an integral role in decision-making of an agent so, more time should be invested to figure out the best heuristics function and apply it.

Acknowledgement: For this assignment, my friend Romain Bazin from ECSE – 526 helped me a lot on how to best design a program in an efficient way and helped me clear some of my doubts regarding the algorithm.