CS F407 – Artificial Intelligence

Report on Programming Assignment #2

Due on November 14, 2023 at 9pm

Professor Dr. Sujith Thomas

Arnav Goyal

2021A7PS2596G

Submitted on November 14, 2023

Table of Contents

1. Comparison between Evaluation Functions

This section contains the results and details of the two different evaluation functions used and comparison between them.

1.1 Description of Evaluation Functions

The function then counts the number of sets of consecutive pieces for both the player and the opponent. A set can be either two (twos), three (threes), or four (fours) consecutive pieces of the same colour.

Evaluation Function – 1

The first evaluation function counts the number of sets of consecutive pieces for both the player and the opponent, focusing on sets of three (threes) and four (fours) consecutive pieces.

If the player has at least one set of four consecutive pieces, the function returns a large positive value, signifying a winning condition for the player. If the opponent has at least one set of four it returns a large negative value, signifying a losing condition for the player.

If the game is neither in a winning nor losing state, the function then calculates an evaluation score based exclusively on the number of sets of three consecutive pieces. The evaluation score is the difference between the number of triplets the player has (my_threes) and the number of triplets the opponent has (opp_threes).

Evaluation Function – 2

In case of winning or losing state, this function does the same as evaluation function – 1 by returning a large positive and negative value respectively.

If neither player is in a winning or losing state, the function calculates an evaluation score based on the number of sets of two and three consecutive pieces.

It assigns weights to these sets: each set of three is worth 10 points, and each set of two is worth 2 points. The player's score is increased by their sets (my_threes * 10 + my_twos * 2) and decreased by the opponent's sets (opp_threes * 20 + opp_twos * 2). Notably, the opponent's sets of three are given a higher weight (20) than the player's sets of three (10), indicating that the function prioritizes blocking the opponent's potential wins more heavily.

1.2 Comparison based on the Number of Games Won

Depth	Wins (W)	Draws (D)	Losses (L)
3	18	2	30
4	21	4	25
5	23	3	24

Table 1 :- The outcomes of 50 games played at particular depth (3,4 and 5) using alpha-beta pruning with evaluation function - 1.

Depth	Wins (W)	Draws (D)	Losses (L)
3	32	5	13
4	38	4	8
5	42	4	4

Table 2 :- The outcomes of 50 games played at particular depth (3,4 and 5) using alpha-beta pruning with evaluation function - 2.

We can observe from the above data of 50 games that the evaluation function-2 which takes into account weighted triplets and pairs performs significantly better than evaluation function-1 which only takes into account difference in triplets between the player and the opponent.

In Connect-4, an evaluation function that counts both triplets (three connected pieces) and pairs (two connected pieces) and assigns weight to them tends to be more effective than one that simply counts triplets for several reasons:

- **Strategic Depth**: By considering both triplets and pairs, the evaluation function gains more depth in understanding the game state. Triplets are significant because they are one step away from a winning line of four, but pairs are also important as they are the building blocks for creating triplets and, eventually, a winning line.

- **Flexibility in Game Play**: An evaluation function that accounts for both pairs and triplets can better adapt to different situations on the board. It can identify potential threats and opportunities earlier than one that only considers triplets. This allows for more strategic moves, such as blocking an opponent's potential triplet or extending one's own pair into a triplet.

- **Weighted Analysis**: Assigning different weights to triplets and pairs allows for a nuanced understanding of the board's state. A triplet might be weighted more heavily because it's closer to achieving a connect-four, but a pair could also be given significant weight if it lies in a strategically advantageous position (like the centre columns).

1.3 Comparison based on Average no. of Moves before Each Win

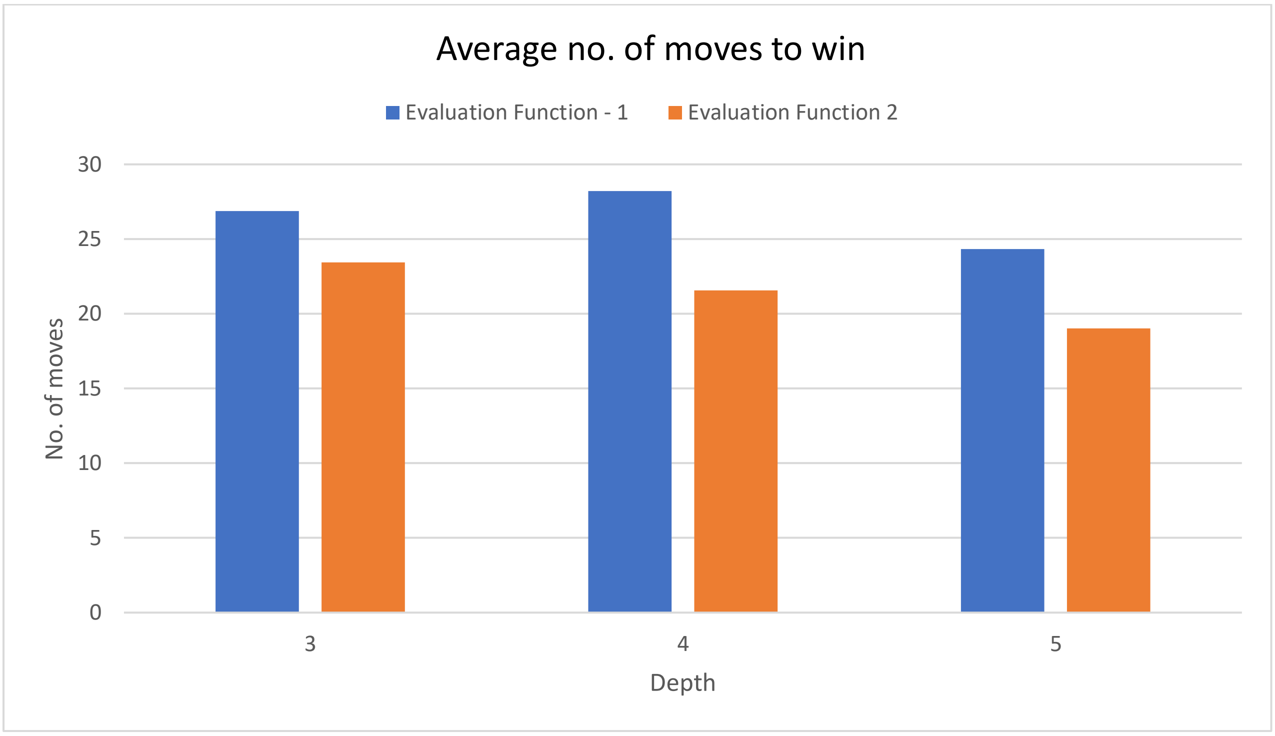

Figure 1 :- Average no. of moves before each win over 50 games played at particular depth (3,4 and 5) using alpha-beta pruning with evaluation function – 1 and 2.

Depth	Average no. of moves to win	
	Evaluation Function - 1	Evaluation Function - 2
3	26.891	23.451
4	28.222	21.565
5	24.333	19

Table 3 :- Numeric data of average no. of moves before each win over 50 games played at various depths using alpha-beta pruning with evaluation function – 1 and 2.

We can observe significant reduction in average no. of moves before each win in case of evaluation function – 2 when compared to evaluation function – 1 in case of depths 4 and 5 and marginal improvement in case of depth 3.

Depth	Improvement(%)
3	12.79
4	23.59
5	21.92

Table 4:- Improvement in no. of moves to win in case of evaluation function – 2 over evaluation function - 1.

This is because of the following reasons: -

- **Early Threat Identification and Strategic Play**: The first evaluation function gives weight to both pairs and triplets and assigns them appropriate weights. By considering pairs, the AI can identify potential threats and opportunities earlier in the game. This allows the AI to make more strategic moves, such as setting up future triplets or blocking the opponent's potential triplets, leading to a quicker path to victory.

- **Focus on Board Control**: By considering pairs, the first evaluation function is more adept at controlling key areas of the board, particularly the centre columns where multiple connect-four opportunities can be created. This control can lead to quicker victories as it opens more opportunities for the AI to win.

2. Move Ordering Heuristic

2.1 Introduction

This section contains the result of application of Move-Ordering heuristic on Minimax with alpha-beta pruning and comparing its advantages in terms of running time of the algorithm, reduction in recursive calls needed to beat the myopic player.

Move ordering is a heuristic used in game-playing algorithms like Minimax with alpha-beta pruning, designed to improve their efficiency.

2.2 Move Ordering Heuristic Used - Prioritising Central Moves

The first move ordering implemented in Connect-4 prioritises moves towards the centre of the board. These moves are often more valuable because they offer more opportunities for connecting four pieces including horizontal apart from vertical and diagonal.

On the other hand, moves on the edge of the board that is columns 0,1,5,6 offer less opportunities and have very limited directions in which four pieces can be connected.

2.2.1 Running Time Comparison

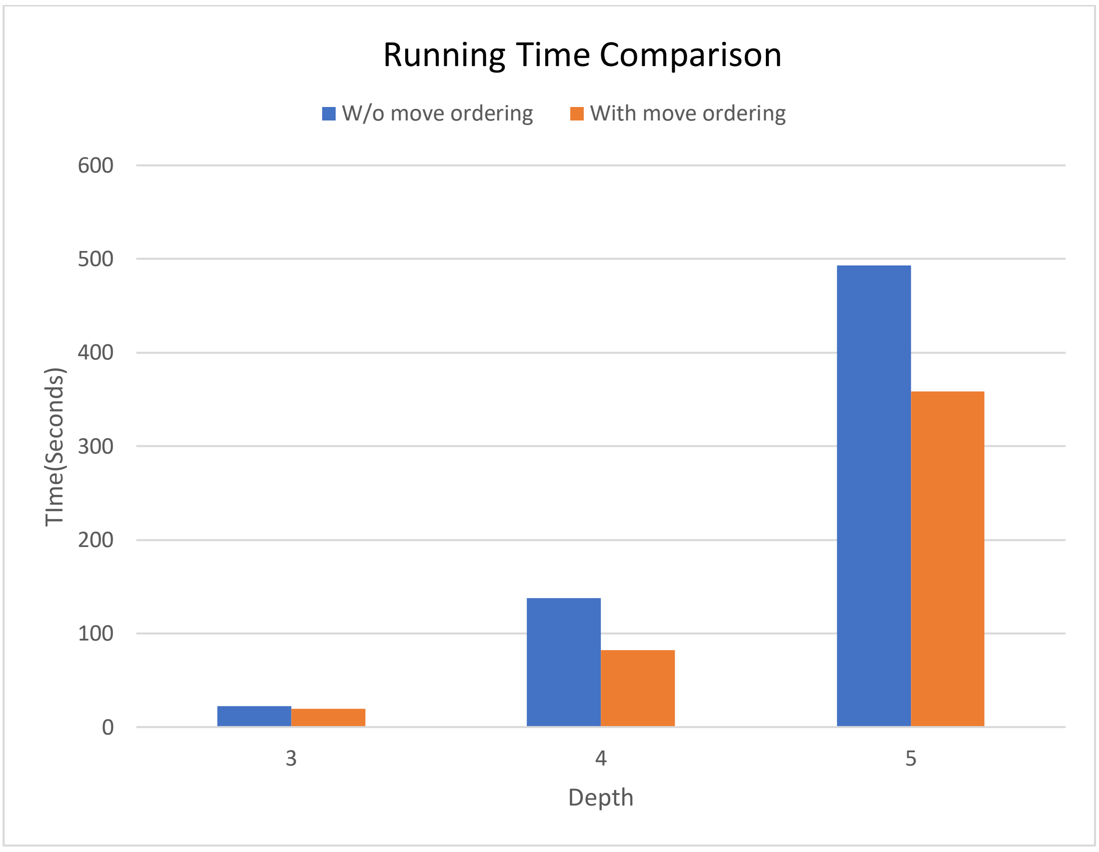

Figure 2 :- Comparison of running times of the game tree search algorithm with and without move ordering.

Depth	Running Time (s)	
	W/o Move Ordering	With Move Ordering
3	22.11	19.54
4	138.04	82.5
5	493.15	358.61

Table 5 :- The comparison of running time of 50 games played at particular depth (3,4 and 5) with and without move ordering.

- A significant reduction in running time was observed when move ordering was implemented. This is because the core idea behind this heuristic is to evaluate the most promising moves first. This prioritization increases the likelihood of early pruning in the alpha-beta process, thus reducing the number of nodes the algorithm needs to examine.

- The execution time improvement in case of depths 4 and 5 was 40.2% and 27.38% respectively. The improvement in execution time becomes necessary as exponentially more computational resource and time is required with increase in depth.

2.2.2 Average no. of Recursive Calls Comparison

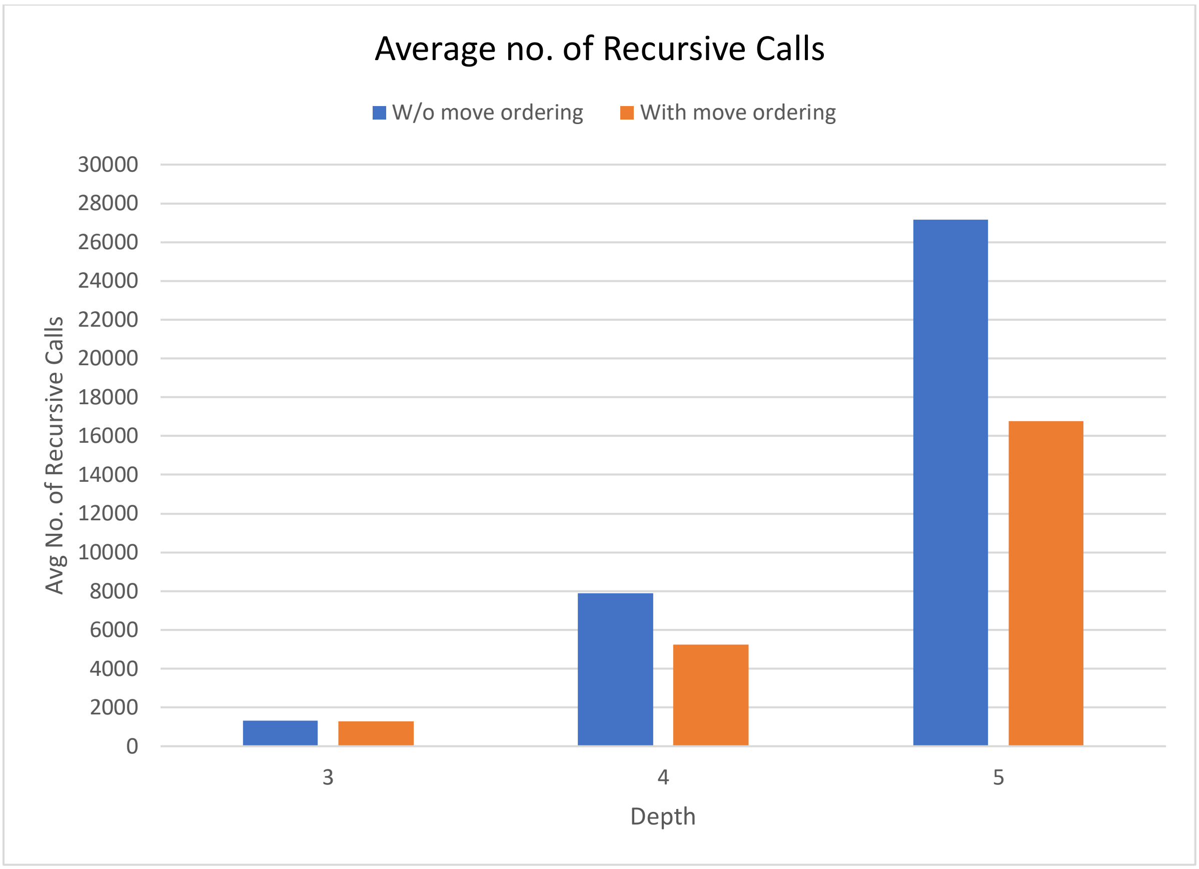

Figure 3 :- Comparison of average no. of rec. calls of the game tree search algorithm at various depths with and without move ordering.

Depth	Average no. of Recursive Calls	
	W/o Move Ordering	With Move Ordering
3	1312.44	1270.06
4	7889.6	5237.4
5	27172.06	16775.82

Table 6 :- The comparison of avg. no. of recursive calls of 50 games played at particular depth (3,4 and 5) with and without move ordering.

- A marginal improvement in average no. of recursive calls was seen for depth 3 whereas for depths 4 and 5, there was significant improvement was seen with nearly 33.6% and 32.86% respectively.

- This is because move-ordering increases the chance of early pruning in the alpha-beta minimax algorithm and hence, the no. of recursive calls is reduced as the no. of nodes the algorithm needs to evaluate is significantly reduced.

3. Comparison between different Cutoff Depths

This section contains the results and details of the comparison between various cutoff depths used in the game tree search algorithm used for the AI, that is the minimax algorithm with alpha-beta heuristic.

3.1 Performance Comparison in terms of Win Percentage

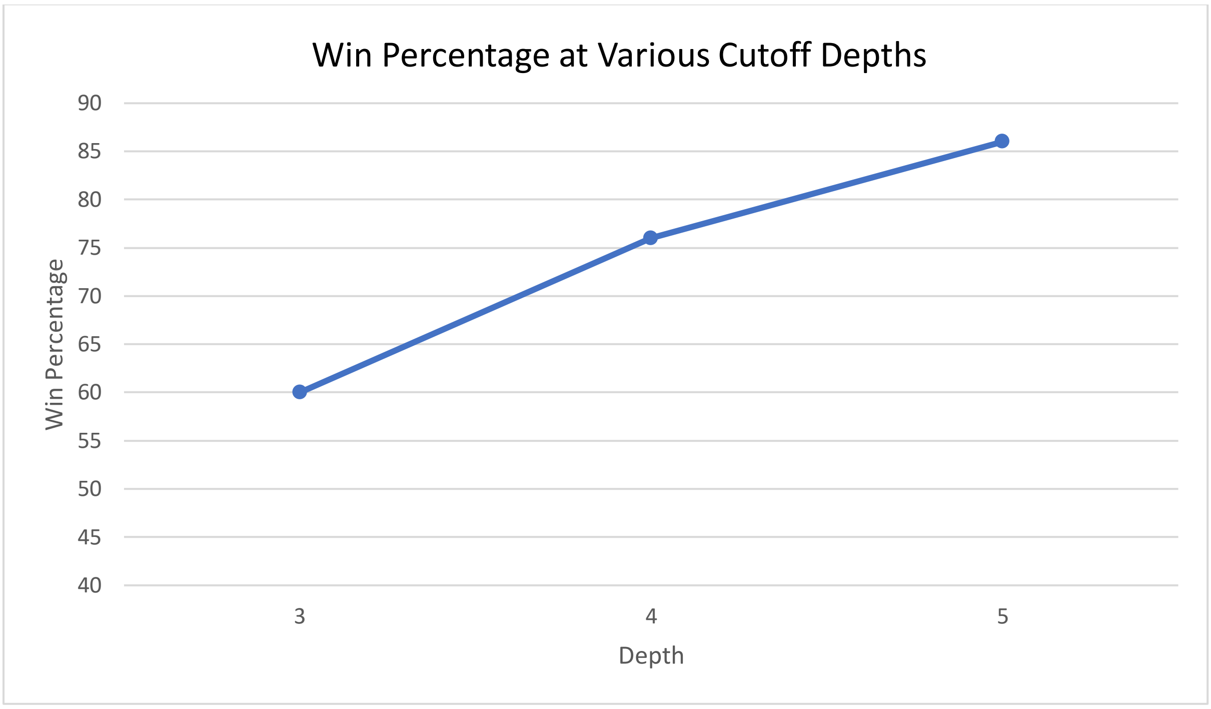

Figure 4 :- Win Percentage in over 50 games played at particular depth (3,4 and 5) using alpha-beta pruning and move ordering.

Depth	Wins (W)	Draws (D)	Losses (L)
3	30	3	17
4	38	5	7
5	43	4	3

Table 7 :- No. of Wins, Draws and Losses in over 50 games played at particular depth (3,4 and 5) using alpha-beta pruning and move ordering heuristic.

The win percentage of the GameTree player increases with increasing depth against the myopic player. This is because a greater cut-off depth allows for :-

- **Greater Look-Ahead Capability**: A higher cutoff depth allows the algorithm to look further ahead in the game. At depth 5, the algorithm evaluates the consequences of moves over a longer sequence than at depth 3. This deeper analysis helps in identifying better moves and strategies, as it considers a wider range of possibilities and outcomes. It can foresee and plan for more of the opponent's potential responses, making the AI more strategic.

- **Improved Decision-Making in Complex Situations**: Connect-4 games can have complex states where the best move isn't immediately obvious. A higher depth allows for a more thorough evaluation of complex board states, helping the AI to identify the most advantageous moves even in intricate situations leading to optimal decisions.

- **Strategic Long-Term Planning**: A higher depth facilitates advanced strategic planning, allowing the AI to set up traps, create multiple threats, and effectively counter the opponent's strategies, aligning immediate moves with a broader game plan.

3.2 Performance Comparison in terms of Average no. of Moves

Depth	Average no. of moves to win
3	24.54
4	24.21
5	22.34

Table 8 :- Average no. of moves before each win over 50 games played at various depths using alpha-beta pruning and move ordering.

- We can observe a marginal improvement in terms of average no. of moves before each win when comparing depths 3 and 5.

- Increasing the cut-off depth improves the AI's ability to analyse and anticipate future game states. This deeper analysis leads to more strategic and effective moves, as the AI can consider the consequences of moves further ahead in the game. This strategic foresight typically results in a more efficient path to victory, reducing the average number of moves before each win.