Impact of Depth of the Minimax Algorithm with Alpha-Beta Pruning on the Strength of Artificial Intelligence Against Humans in Connect 4 in a Square

*

1st Ahmad Shalaby
Computer Science CS 580
George Mason University
Fairfax, United States of America
ashalaby@gmu.edu

***Abstract*—Games have always interested AI researchers due to how difficult they are to program and implement [1]. For many games, it is impossible for the algorithm to calculate the optimal solution due to nearly infinite combinations of pieces and actions [1]. This paper utilizes the minimax algorithm with alpha-beta pruning to create an AI with five levels of difficulty to play connect four in a square. It was found that an artificial intelligence utilizing a depth level of two for the minimax algorithm had the greatest success against the human player. These findings suggest that a depth level of 2 for games such as connect four in a square is enough for difficulty, and developers should look at other avenues, such as introducing randomness or modifying the heuristic functions, to modify difficulty in their games.**

***Index Terms*—Minimax Algorithm, Alpha-Beta Pruning, Adversarial Agents**

I. Introduction

The game is an adversarial search in a multi-agent environment [2]. Also known as games, adversarial search problems occur when agents must plan ahead against other agents who are also planning ahead and respond to them [1, 3]. Examples of algorithms used in artificial intelligence board games include greedy, alpha-beta pruning, principal variation search, and monte Carlo tree search [3] Connect four is an adversarial, finite zero-sum game with perfect information [3]. This means that when one agent wins the other loses, the board is deterministic with no probability involved, and the environment is completely observable [1] The algorithm used by the agent in this experiment was the minimax algorithm at different depths alongside alpha-beta pruning to speed up the algorithm. The minimax algorithm has been previously used in games such as chess, backgammon, and connect 4 [4]. This report begins with a background explaining this algorithm, alongside the rules to play connect 4 in a square. It then goes into the specific approach and application used in this paper, ending with the results of the experiment and conclusions by the author.

II. Background

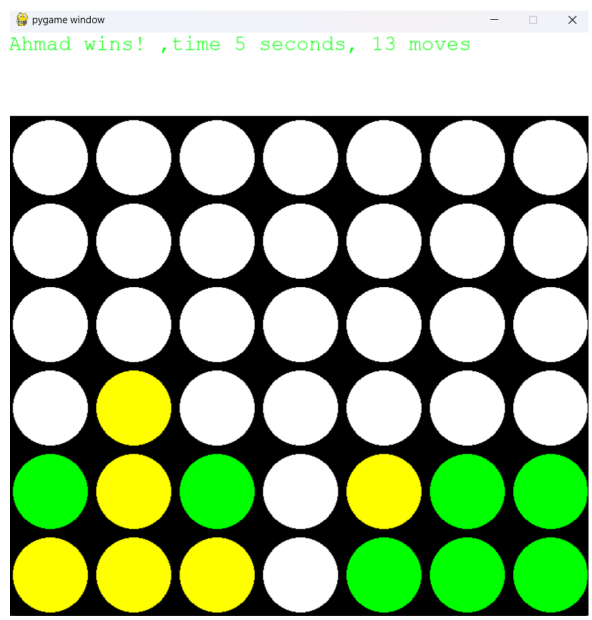

Fig. 1. Winning board state in Connect 4 for the green player pieces.

A. Connect 4 in a Square

Connect 4 in a square is played on a board which contains 7 vertical columns and 6 rows of empty squares. Players take turns placing their pieces on the bottom of each column. This game is like connect 4, however, the winning condition is that the 4 circles placed must form a square of the same pieces.

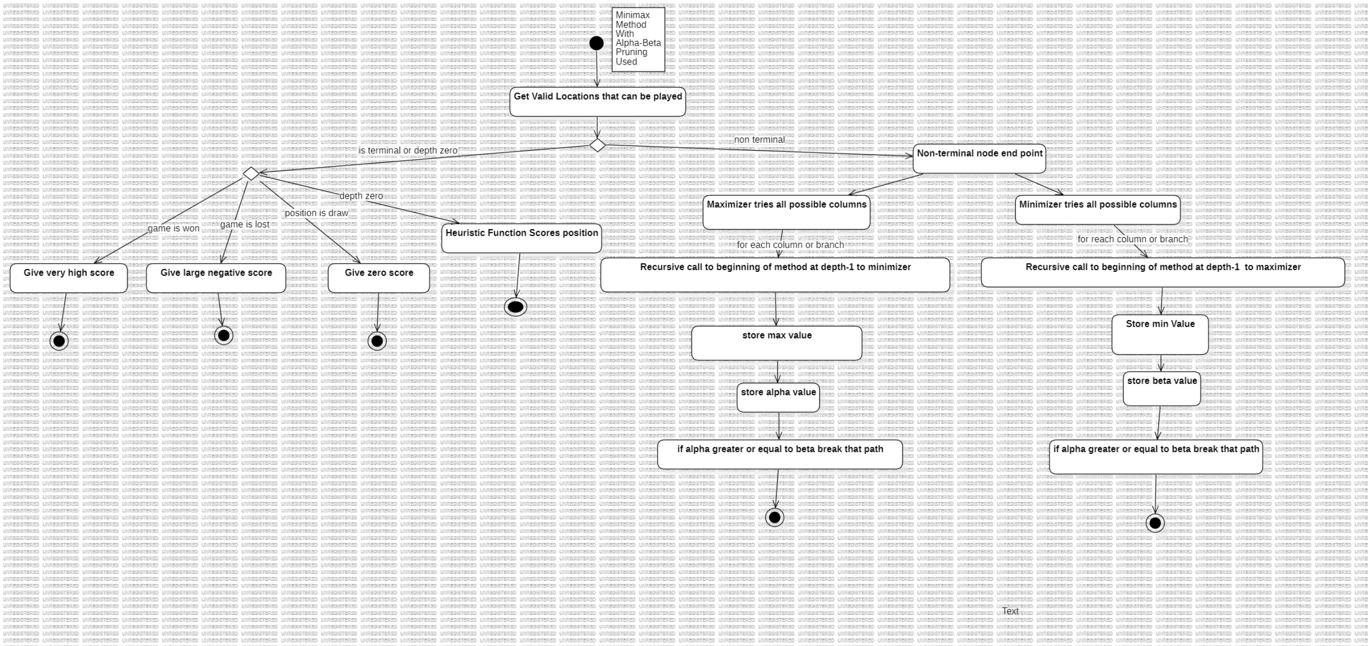

Fig. 2. Minimax algorithm used for the algorithm to generate minimax values for each possible column in connect four in a square. As shown, the algorithm handles terminal nodes and non-terminal nodes differently. When the function is either at a terminal node or when depth has reached zero, it evaluates the score of the function with a heuristic scoring function. The winning end state is given a very high value, which promotes the algorithm to pick it, while the losing state is very low suggesting the algorithm to not lose when possible. Non-terminal nodes differ in whether the player is the MAX or MIN player. As already mentioned previously, the maximizer will go through recursively until reaching an end state or pruning.

B. The Minimax Algorithm and Alpha-Beta Pruning

For the artificial intelligence to play connect 4 in a square effectively, the minimax algorithm alongside alpha-beta pruning is applied. The minimax algorithm was proposed by John con Neumann in 1928 [4].

The minimax algorithm computes the minimax values for every node/board state, and then backtracks to find the best move [4]. The minimax algorithm is made up of two players, one defined as the MAX player and the other the MIN [1]. Max makes moves to get the highest score, while MIN moves to lower the score [4]. In any given move the minimax algorithm picks from between the children nodes (new moves) to either maximize or minimize the score based on who is playing [1]. Recursion is used to proceed down the leaves of a tree for the required depth [1].

One weakness of the minimax method is the long runtime, to combat this, alpha-beta pruning as well as heuristics are applied [4] Heuristics functions give scores/utility to positions so that the minimax algorithm does not have to do a complete search to the end of a game to see which path/move to take [1,4]. Heuristic functions can provide scores based on a wide range of possibilities and are very important to the efficiency of the minimax algorithm [4]. In relation to Connect four in a square, scores given can include how the pieces fit together on the board, as well as the location of pieces [4].

Alpha-Beta pruning is a method which “prunes” useless nodes on the search tree to accelerate the search [3]. Pruning means ignoring portions of the search tree which are known to have no impact on the final choice which will be made [1]. Alpha-Beta pruning is essentially an optimization of the minimax algorithm, on the same depth alpha-beta pruning takes much less time than minimax to generate game states [2]. The pruning condition is when alpha is greater than Beta [3]. Alpha is the score of the max layers while Beta is the score of the min layers in the minimax tree [3]. When alpha is greater than beta, it informs us that the child nodes will never be accessed, this is because the action is going to be worse than other already discovered actions [3]

III. Proposed Approach

This paper utilizes a basic minimax algorithm alongside alpha beta pruning to speed up the algorithm at higher depths. The human player plays against the Artificial Intelligence opponent seven times across five different difficulties, with each difficulty being the depth of the minimax algorithm.

The Heuristic function used for this minimax algorithm gave points based on the position of the pieces on the board in relation to the board and each other after a piece was placed. The heuristic function scored every edge piece as minus 3 points to promote the minimax algorithm to not choose to place the pieces on the edge of the board as that limits the amount of squares/win conditions that can be created. Placing two pieces next to each other granted 25 points and placing two pieces in a winnable “three in a row” L shape granted 50 points (on top of the 50 points generated from the present “two in a rows”).

Lastly, a “pick best move” function was called that uses the minimax algorithm. Because there is only one AI, this pick best move function will always call for the maximum value and played that.

IV. Expiremental Results

In order to not repeat the games, the human player must go first and start with a different move in each game. In this paper, the player starts at column 1 in game 1, and ends on column 7 in game 7. The human player plays seven games across 5 different artificial intelligence difficulties, each difficulty representing the depth of the minimax algorithm. This was to see the effect of depth on how powerful the minimax algorithm was at playing connect 4 in a square. Difficulty zero was introduced as a control parameter in the experiment, which had the artificial intelligence play completely random possible moves.

Level	Game 1	Game 2	Game 3	Game 4	Game 5	Game 6	Game 7
Level 0	player	Player	Player	Player	Player	Player	Player
Level 1	player	Player	Player	Player	Player	Player	Player
Level 2	Ai	Ai	Ai	Draw	Ai	Draw	Draw
Level 3	Draw	Draw	Ai	Draw	Draw	Draw	Draw
Level 4	Draw	Draw	Draw	Draw	Draw	Draw	Draw
Level 5	Draw	Draw	Draw	Draw	Draw	Draw	Draw

TABLE I

Raw data table: table indicates which games were won by the human player or the AI algorithm. Games ending in a draw were labeled as draw.

Level	Player Winrate (%)	Moves Played	Time (s)
0	100	9.3	4.7
1	100	12.4	6.3
2	21.4	32.3	25.1
3	42.9	37.1	23.1
4	50	42	15.6
5	50	42	24.9

TABLE II

Processed Data Table

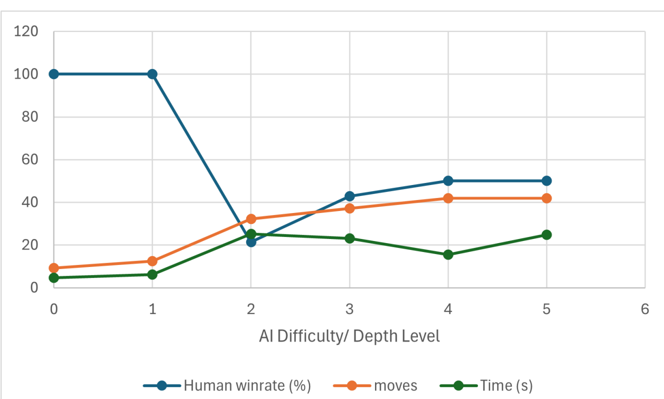

Fig. 3. Scatter plot with separate lines to compare the human win rate, number of moves played, as well as time between the human player and AI at different difficulties.

Processed data table shows the average across the seven games at each AI level, as shown in the table. A win rate of 100 meant that the player won every game, while 0 would mean the player lost every game. The human win rate was lowest at depth equal to two, and highest at depth 1 and against random play. The moves required to reach and end state increased as the level increased, while the time to play a game did increase after level 1 and did not have a noticeable change after that, despite a small dip at level 4.

V. Conclusions

Connect four is a solved game [2]. This suggests that connect 4 in a square, and even simpler game, can also be solved. For this reason, I believe that the reason why the human player lost against the “easier” artificial intelligence level 2 Moreso than level 5, was because of the nature of the minimax algorithm. The minimax algorithm assumes that the opponent player will play optimally to the level of its depth. At depth level 5, playing optimally ensures that basically no one will ever win, since you would have to think 5 moves in advance to outplay your opponent. In such a small board, it is impossible to set up a win that cannot be stopped within 5 moves, so the algorithm did not even try and instead just maximized the heuristic.

However, at level 2, there are tricks that can be used to outplay an opponent and go for a win. For example, forming an upside-down T shape can result in two separate possible boxes that cannot be blocked by your opponent. This was the method used by the player to get a perfect win rate against the level 1 Ai, however, the level 2 Ai was able to detect this strategy and never allowed it to occur. I noticed that the level 2 AI also attempted this strategy against the human player to some success, while the higher-level AI did not even attempt this strategy, which is believed to be due to assuming the player will play optimally.

There is an interesting dichotomy between artificial intelligence and human intelligence, where a weaker artificial intelligence can perform stronger against a human player than a more powerful one. There is a saying of chess, that a perfectly played game of chess always ends in a draw, which seems to apply here that when a game is solved, the artificial algorithms seem to always push for draws. While a human thinks of conceptual patterns to lead to wins, i.e., forming an upside-down T shape to get two possible victories, the minimax algorithm looks at board states and plays moves to maximize its heuristic. Therefore, to upgrade the algorithm to win against human players, rather than increasing the depth, the heuristics used should be changed. One example is to include traps to create more “aggressive” heuristics, for example, the Ai would gain more points if its pieces were arranged in the upside-down T. Another modification that could be used to manage difficulty is introduce randomness at lower difficulties. However, if the goal of the algorithm is to never lose, then a high depth is appropriate.

According to [3] in “Comparison of Four AI Algorithms in Connect Four,” Monte Carlo tree search algorithm provided the highest win rate playing against other artificial intelligence algorithms [3]. So perhaps other algorithms may provide better win chances for artificial intelligence against humans. One confounding factor was that as the human player played more games, they gained more practice and were able to make less mistakes. Perhaps if the level 2 games were played last, the human player would have had a higher draw rate against them. Furthermore, because the human player also designed the code, they knew how to outplay the level 1 depth Ai granting them easy wins which contributed to them playing hastily against depth level 2. To get a more accurate representation, stronger human players who have not seen how the AI thinks should play against the different levels. '

References

[1] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Upper Saddle River: Prentice-Hall, 2010.

[2] R. Nasa, R. Didwania, S. Maji, and V. Kumar, "Alpha-Beta Pruning in Mini-Max Algorithm –An Optimized Approach for a Connect-4 Game," International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 4, Apr. 2018.

[3] Y. Qiu, Z. Wang and D. Xu, "Comparison of Four AI Algorithms in Connect Four," MEMAT 2022; 2nd International Conference on Mechanical Engineering, Intelligent Manufacturing and Automation Technology, Guilin, China, 2022, pp. 1-5.

[4] X. Kang, Y. Wang, and Y. Hu, "Research on different heuristics for Minimax algorithm insight from connect-4 game," Journal of Intelligent Learning Systems and Applications, vol. 11, no. 02, pp. 15–31, 2019. doi:10.4236/jilsa.2019.112002