AI Agent for Playing the Game of Ultimate Tic Tac Toe

Vraj Patel and Daishak Patel

***Abstract* - Ultimate Tic Tac Toe enhances the classic game by featuring 9 smaller grids within a larger grid. The objective is to win individual regions to claim a row, column, or diagonal in the overall board. This project aimed to develop an AI agent using minimax and alpha-beta pruning techniques to play against novice players effectively. Given the game's high branching factor, standard minimax would struggle due to its exponential time complexity. To optimize performance, we implemented alpha-beta pruning and evaluated three heuristic functions, specifically positional advantage, number of captured regions, and number of free goes offered. Results showed that combining minimax with alpha-beta pruning and the positional advantage heuristic functions resulted in a powerful AI with the highest win rate.**

I. Introduction

Ultimate Tic-Tac-Toe presents a captivating twist on the timeless game of regular Tic Tac Toe, enhancing it with a multi-grid structure that introduces depth and strategy. Players seek to win by conquering smaller Tic-Tac-Toe grids within a larger framework. This dynamic unfolds on a 3x3 grid of grids, where victory entails securing three small grids in a row.

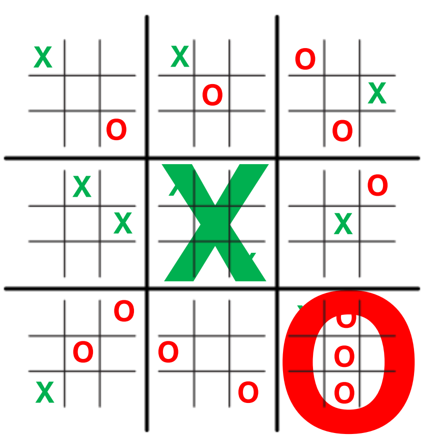

Fig. 1. Ultimate Tic Tac Toe Board [1]

A. Game Setup

1. The game unfolds on a 3x3 grid comprising nine smaller grids, arranged in a 3x3 layout.

2. Each small grid operates exactly like a standard Tic-Tac-Toe, with players placing their marks (X or O) until one emerges victorious or the grid reaches capacity.

3. However, victory in Ultimate Tic-Tac-Toe is attained by dominating the larger grid, achieved by securing three small grids in a row, either horizontally, vertically, or diagonally.

B. Gameplay

1. Players alternate placing their marks in any allowed and vacant square of the large grid.

2. The chosen square dictates the small grid where the opponent must make their subsequent move.

3. For instance, if Player A marks the top-right cell of a small grid, Player B must respond in kind in the corresponding grid of the larger layout.

4. When a player captures a region, their mark is displayed in the corresponding cell of the large grid.

5. Should a player's move lead to an already won or full small grid, the opponent gains the freedom to play in any vacant cell of the large grid - this is known as a “free go.”

6. The game progresses until one player secures victory by securing three small grids in a row or until the large grid reaches capacity, in which case the player with most captured regions wins.

7. If the larger board reaches capacity and both players have the same number of captured regions, the game is tied. [1]

C. Objective of the AI System

The objective was to develop an AI system adept at engaging novice players in Ultimate Tic-Tac-Toe, offering a stimulating yet enjoyable experience. Leveraging adversarial search algorithms, particularly the minimax algorithm bolstered by alpha-beta pruning, the AI strives for efficient decision-making while preserving strategic depth. Additional enhancements, including region and position heuristics, boost the AI's performance.

II. Description of the AI System

The AI system's foundation lies in the minimax algorithm, implemented as a recursive function in the code with a depth limit in case terminal states (win, lose, or tie) are not reached, which is common throughout most of the game [2]. When the depth limit is reached, the resulting board configuration is passed to an evaluation function that gives the board a score based on different heuristic functions. A positive score benefits the computer which is assigned to be the maximizing player, and a negative score benefits the human player. This evaluation function is the key to decision-making through the simulation of alternate game states.

Six versions of the AI agent were designed for the experiment. Each subsequent agent has additional, more advanced features implemented.

1. ***Random Agent (R)*** – this agent would randomly select from the legal moves, thus serving as the control group with which to compare the other versions.
2. ***Minimax Agent (M)*** – this agent used raw minimax, with a depth limit cut off at 3 which was the greatest depth it could reach. No heuristics were used.
3. ***Alpha Beta Pruning Agent (AB)*** – this agent incorporates alpha beta pruning to the minimax algorithm. Can achieve a depth of 5 but does not make use of any heuristics.
4. ***Captured Regions Heuristic Agent (CR)*** **–** This is the first agent that introduces a heuristic in the evaluation function at the depth cut off. The captured regions heuristic counts the number of smaller regions within the larger layout that have been captured by a player.
5. ***Positional Advantage Heuristic Agent (PA)*** - This agent adds another heuristic that counts the number of opportunities that a board configuration gives a particular player. Opportunities refer to when a player has two marks and only one more is needed to capture that region.
6. ***Free Goes Heuristic Agent (FG)*** **-** The final agent includes a third heuristic – the number of free goes offered to the opponent in reaching a particular board state.

Note: These agents build upon previous agents' features. Therefore, the last agent includes all three heuristics combined with alpha beta pruning.

To summarize the AI system, the computational complexity inherent in Ultimate Tic-Tac-Toe, was mitigated by integrating alpha-beta pruning, significantly streamlining the evaluation process. Additionally, the higher AI versions incorporate region and position heuristics to refine decision-making, prioritizing moves based on strategic significance. These features were predicted to collectively enhance performance in terms of move time and depth of exploration, ensuring a challenging yet balanced gaming experience.

III. Experimental Results

The two primary measures used to evaluate the AI agents were win rate and response time, with an emphasis on win rate. Fifty games were played with each version of the AI by novice players with a basic understanding of strategy and the average response time per move and the number of wins were recorded. The results are summarized in the following table.

AI Agent	R	M	AB	CR	PA	FG
Wins out of 50	2	5	7	14	41	34
Win Ratio	4%	10%	14%	28%	82%	68%
Average Response Time (s)	0.8 s	6.7 s	3.3 s	3.6 s	4.5 s	5.1 s
Average number of nodes searched	156	504,632	204,126	190,045	210,121	196,447

Fig. 2. Summary of experimental results for all six version of the AI Agent.

IV. Discussion and Analysis

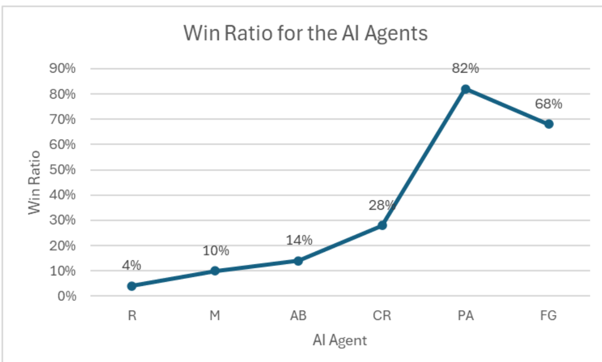

Fig. 3. Line graph depicting the trend in win ratios as more advanced AI agents were introduced.

The results showed that the fifth AI agent, which included the positional advantage and captured regions heuristics was optimal with a staggering 82% win ratio. The results were for the most part as hypothesized – introducing additional features such as alpha beta pruning and heuristics increased the AI's difficulty level, with exception the free goes heuristic. The reason that the alpha beta pruning agent was largely unsuccessful was despite searching at a greater depth, it still could not reach the terminal states in the game and thus ended up making random moves, albeit in the end game where it could effectively reach the winning configurations. The introduction of the positional advantage heuristic saw the greatest jump in winning rate as it made the AI capable of blocking the opponent's opportunities to capture regions.

A. Performance Evaluation

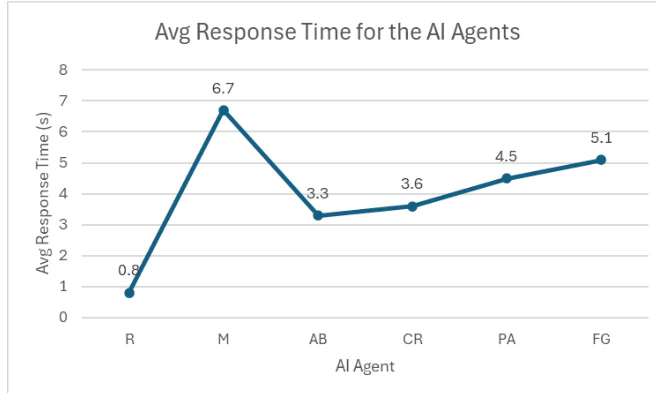

Fig. 4. Line graph depicting the trend in average response time per move with increasingly advanced agents.

Regarding response times, the random agent was the fastest since it implemented no search techniques but was supremely easy to beat. All subsequent agents based off the minimax algorithm had an exponential time complexity of $O(b^d)$ where the branching factor b was at most 9 for regular moves and at most 81 for free goes (decreased as game progressed). The maximum depth was gradually increased from 0 to 3 for the raw minimax agent and from 0 to 5 for the four agents that used alpha beta pruning, since excessive depth at the beginning stages of the game is unnecessary.

The raw minimax agent, which achieved only a max depth of 3, was the slowest at 6.7 seconds and alpha beta pruning was able to reduce response time to 3.3 seconds by cutting down the number of nodes searched by half. The three subsequent heuristic agents were built from the alpha beta pruning agent (and thus searched comparable number of nodes on average) but involved additional computations for each additional heuristic, explaining the linear increase in response time with each additional heuristic computed.

B. Discussion of Surprising Discoveries

An unexpected revelation was the superior performance of the positional advantage heuristic in refining the AI's gameplay than the agent that also included the free goes heuristic. Despite initial expectations favoring the integration of all three heuristics, it became evident that the position heuristic and captured regions heuristics alone sufficed to achieve optimal results. A reason the free goes heuristic proved detrimental to the win ratio could be that the minimax algorithm already accounted for its underlying idea since the algorithm iterates through all of the moves both minimizing player and maximizing player can make, including the increased number of moves resulting from free goes. This underscores the importance of tailoring specific aspects of the AI system to maximize performance effectively.

C. Limitations and Future Work

A notable challenge encountered during the investigation was the absence of established heuristic functions tailored for Ultimate Tic Tac Toe [3]. Developing heuristic functions that accurately captured the game's unique dynamics and strategic nuances proved to be a significant hurdle. Overcoming this challenge necessitated iterative refinement and experimentation to devise heuristics capable of guiding the AI's decision-making effectively.

Future research endeavors could explore the integration of advanced methodologies such as Monte Carlo Simulation to further enhance the AI's gameplay capabilities [3]. MCS offers promising prospects for improving decision-making in complex environments by simulating numerous outcomes and iteratively refining search strategies based on probability estimates. Additionally, ongoing refinement of heuristic functions tailored specifically for Ultimate Tic Tac Toe could unlock additional performance enhancements, paving the way for more formidable AI opponents.

V. CONCLUSION

The investigation revealed that the AI system's optimal performance in Ultimate Tic Tac Toe was achieved when equipped with the positional advantage and captured regions heuristic. Due to the high branching factor and inability to reach the terminal states in a minimax search even with alpha beta pruning, the introduction of various heuristic evaluation functions was indispensable. Of the three heuristics used, the positional advantage heuristic proved to be the most significant leading to the highest win ratio of 82% out of all of the agents when combined with alpha beta pruning and the captured regions heuristic. Contrary to expectations, integrating the free goes heuristics decreased the win ratio, highlighting the dominance of the position heuristic in enhancing decision-making accuracy for the AI system. This project resulted in the creation of an intuitive console-based Ultimate Tic Tac Toe game written in the Python language that was easy to learn and exciting for novice players to play with. To create more powerful AI opponents suitable for advanced players, other methods such as Monte Carlo simulation could be investigated in the future since they do not rely on heuristics but rather probabilities.

Fig. 4. The user interface for the Python console-based game.

REFERENCES

[1] “Ultimate tic-tac-toe,” *The Game Gal*, Sep. 2018.https://www.thegamegal.com/2018/09/01/ultimate-tic-tac-toe/ (accessed Apr. 7, 2024).

[2] in, “The Minimax Algorithm 101,” *Japlslounge.com*, 2020.https://japlslounge.com/posts/articles/Minimax_Blog/1.htm (accessed Apr. 14, 2024).

[3] Gila, O., “Ultimate tic-tac-toe,” *Wikipedia*, Oct. 18, 2023. https://en.wikipedia.org/wiki/Ultimate_tic-tac-toe (accessed Apr. 10, 2024).