Ultimate Tic-Tac-Toe Bot Techniques

Brycen Addison and **Michael Peeler** and **Chris Alvin**
Furman University
chris.alvin@furman.edu

Abstract

In this work, we consider the game Ultimate Tic-Tac-Toe. We apply different bot-based strategies for play and compare their effectiveness, including: random play, a heat-map approach, Monte Carlo Tree Search, and functionality similar to AlphaZero. Our work reveals another game that AlphaZero crushes the competition.

Introduction. The classic child's game Tic-Tac-Toe is a turn-based two-player game played on a 3×3 board. Ultimate Tic-Tac-Toe (UTTT) extends Tic-Tac-Toe by defining each cell of the 3×3 UTTT game board to be an independent, standard Tic-Tac-Toe game as shown in Figure 1. The goal of UTTT is similar to the original game in that a player must take three cells in a row. However, to claim a cell on the UTTT board, a player must win the corresponding subgame of standard Tic-Tac-Toe. Play in UTTT is turn-based, but a player does not arbitrarily choose where they play; play is determined by your opponent's previous move. For example, if your opponent plays in the middle-left cell on the top-left board (labeled as 4_1 in Figure 1), your next move must be an open cell somewhere in the middle-left board (those cells labeled with subscript 4 in Figure 1). Choosing to then play in cell 9_4 in Figure 1 means your opponent's next move must be made in the bottom-right board.

Experimental Setup. We implemented several techniques for competitive bots. We used a heatmap approach in which the heatmap in Figure 2 was applied to the UTTT board as well as the mini-boards. We also implemented Monte Carlo Tree Search (MCTS) running on a tenth of a second and a full second of search time per move. Last, we implemented an AlphaZero-inspired (Nair 2022) bot built with a convolutional neural network (CNN); the CNN was trained for 24 hours on a dataset generated using MCTS. We executed an experiment in which each bot played against each other bot 10000 times; total wins, draws, and losses for each bot's 50000 games are shown in Figure 3.

Experimental Analyses. With a typical game lasting at least 30 moves, the MCTS cannot reach a depth that vastly outperforms random moves. Heatmaps proved more effective in comparison, and could be used to aid the machine learning process. The AlphaZero bot outperformed all other agents (Figure 3), winning 88.00% of its matches; a greater win percentage may be achieved with further optimization or extended training time.

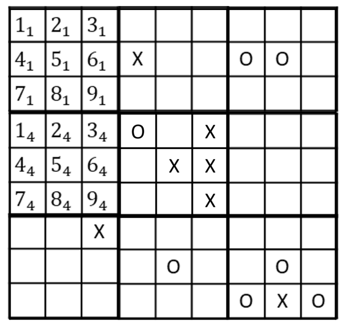

Figure 1: A UTTT board with embedded Tic-Tac-Toe 'mini-boards'.

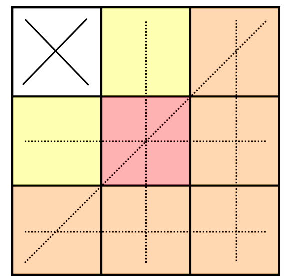

Figure 2: An example heatmap for a board with $\times$ played in the top-left corner. Dotted lines represent possible paths to victory: yellow (1 path), orange (2 paths), and red (3 paths). The cell with the most paths is chosen.

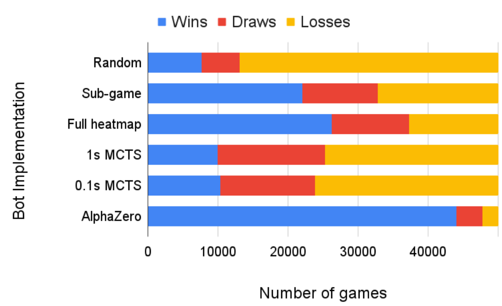

Figure 3: Bot wins, draws, and losses over 50000 trials (10000 for each of 5 bot opponents).

Acknowledgements. This work was supported by The Furman Advantage.

References

Nair, S. 2022. Simple Alpha Zero. https://web.stanford.edu/~surag/posts/alphazero.html.