Experiment 1: Analysis of Tree Size Impact on MCTS Performance

Introduction

The experiment explores the impact of tree size on the performance of Monte Carlo Tree Search (MCTS) in Ultimate Tic-Tac-Toe.

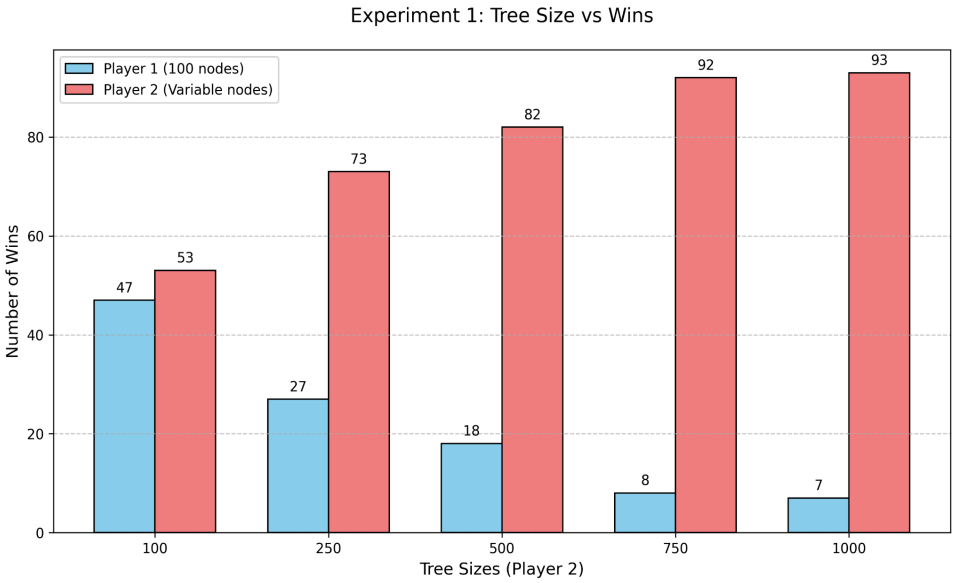
Methodology

- Player 1 (Baseline): Fixed at 100 nodes per tree
- Player 2 (Variable): Tested at 100, 250, 500, 750, and 1000 nodes per tree
- Games Played: 100 per configuration
- Metrics: Number of wins for each player

Results

- Tree Size: 100 nodes
  - Player 1: 47 wins vs. Player 2: 53 wins
  - Computation time: 198.24 seconds
- Tree Size: 250 nodes
  - Player 1: 27 wins vs. Player 2: 73 wins
  - Computation time: 327.27 seconds
- Tree Size: 500 nodes
  - Player 1: 18 wins vs. Player 2: 82 wins
  - Computation time: 565.49 seconds
- Tree Size: 750 nodes
  - Player 1: 8 wins vs. Player 2: 92 wins
  - Computation time: 765.33 seconds
- Tree Size: 1000 nodes
  - Player 1: 7 wins vs. Player 2: 93 wins
  - Computation time: 967.72 seconds

Visual Plot



Analysis

- Strategic Effectiveness
    - Initial Tree Size (100 nodes):
        - Player 1 earns 47% wins and Player 2 achieves 53%, demonstrates the relative balance between the two players, which is intended and expected.
    - Impact of Increasing Tree Size:
        - At 250 nodes, Player 2's win rate increases significantly (+20%) and at 750 nodes, Player 2 continues to perform better, with win rates rising to 92%.
    - Diminishing Returns
        - The increase from 750 to 1000 nodes results in only a 1% improvement in Player 2's win rate, so the time and computing cost might not be worth it.

Computational Cost

The execution time scales linearly with tree size, with the time required for 1000 nodes nearly five times that for 100 nodes. The steep increase in computational cost beyond 750 nodes yields relatively minimal performance gains, so it might not be worth it.

Conclusion

The experiment reveals a clear relationship between tree size and MCTS performance, which is that larger trees lead to better strategic decisions. If a developer was implementing MCTS for a similar use case, it could be worthwhile to save computational costs by determining the optimal tree size, which in this case, might be 750 nodes or even a little less.