Experiment 3: Time as a Constraint

Introduction

The experiment explores the impact of using time as a limiting factor in Monte Carlo Tree Search (MCTS). Instead of fixing the number of nodes in the search tree, the algorithm is allowed to grow the tree as long as it remains within a given time budget.

Methodology

- Time Budgets: 0.1s, 0.5s, 1.0s, and 1.5s
- Implementations Tested: Vanilla MCTS (random rollout) and Modified MCTS (heuristic rollout)
- Metrics
  - Win counts for each implementation
  - Average tree size (number of nodes) generated within the time budget

Results

- Time Limit: 0.1s
  - Vanilla: 46 Wins vs. Modified: 3 Wins
  - Vanilla: 246.4 nodes vs. Modified: 18.6 nodes
- Time Limit: 0.5s
  - Vanilla: 34 Wins vs. Modified: 16 Wins
  - Vanilla: 1328.5 nodes vs. Modified: 119.2 nodes
- Time Limit: 1.0s
  - Vanilla: 43 Wins vs. Modified: 7 Wins
  - Vanilla: 2590.7 nodes vs. Modified: 229.4 nodes
- Time Limit: 1.5s
  - Vanilla: 46 Wins vs. Modified: 4 Wins
  - Vanilla: 3811.5 nodes vs. Modified: 351.5 nodes

Visual Plot
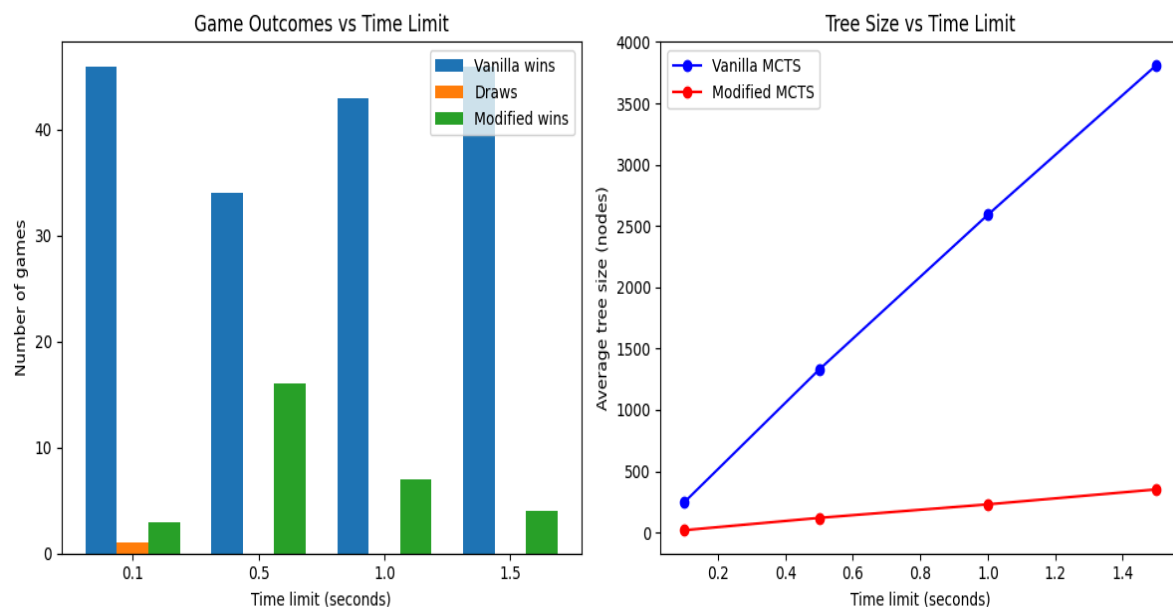
Analysis

- The vanilla MCTS generates larger trees compared to the modified version under all time budgets.
    - The vanilla version performs random rollouts, which are computationally lightweight, allowing it to explore more nodes within the same time.
- The modified MCTS uses heuristic rollouts, which likely involve more computation per action since it often is more accurate compared to vanilla, but it results in smaller trees within the time constraints.

Conclusion

The results show that the modified version of MCTS consistently smaller trees compared to the vanilla version under the tested time constraints, since the heuristic rollouts used in the modified implementation are more expensive than the random rollouts in the vanilla version. In addition, the modified MCTS spends more time evaluating each node, limiting the total number of nodes it can explore within the given time budget. The comparison does not significantly change across varying time limits; the vanilla version consistently generates trees that are roughly 10–15 times larger than those produced by the modified version. However, as the time budget increases, the gap in average tree size narrows slightly, suggesting that the modified MCTS benefits more from additional computation time.