A Gomoku Artifical Intelligence Agent

Jiacheng Hou
jhou013@uottawa.ca

I. Introduction

Gomoku, also called Five in a Row, originates from Japan. It is a two-player, turn-based and zero-sum game played with GO pieces (black and white stones) on a GO board. Black always goes first, and then players alternately place the stone in an empty intersection of the board, and stones can not be moved once placed. The first player who gets five pieces horizontally, vertically or diagonally wins the game.

Gomoku is an exciting game, and it is pretty popular across Japan, Korea and China. In China, we call the game Wuziqi, 'Wu' means five, 'Zi' means piece, and 'Qi' is a board game category in China. Gomoku brought much fun to the author's childhood, although the author did not have a wise strategy at that time.

In 2016, the birth of AlphaGo [1], an Artificial Intelligence (AI) agent, shocked the world by beating a Go world champion. Go is thought to be the most challenging board game globally because there are around 10^{172} possible board positions. It gives the author an insight into the power of AI and inspires the author to build a Gomoku AI player.

This project aims to build a Freestyle Gomoku AI agent utilizing the minimax algorithms with alpha-beta pruning and a greedy approach. The rest of this project proposal is organized as follows. Section II overviews related techniques. Section III presents evaluations among different techniques and parameters.

II. Techniques

The minimax algorithm [2] was initially proposed for zero-sum games with n-players, in which players take alternative or simultaneous decisions. It is widely used in games such as Tic-Tac-Toe, Chess and Gomoku. The minimax algorithm names players as MAX and MIN in a two-player game. The goal of the player MAX is to optimize the payoff, while MIN, on the contrary, minimizes the payoff that MAX can obtain. Minimax works by constructing a state space tree recursively. It continuously traverses possible successors until the leaf node is reached. Then the utility of the leaf node state is backed up as the recursion unwinds. It is worth noting that the algorithm assumes both players play optimally. Besides, it performs a complete depth-first search of the game tree. However, it is a problem for games with a large state space, such as Gomoku. Let b be the branching factor of a state and d be the depth of the state space tree. Then the minimax algorithm has the time complexity $O(b^d)$. In a 15×15 Gomoku board with depth of size 5, the time complexity is around $225^5 = 576650390625$. With the increase of search depth, the time complexity rises exponentially. In reality, it is infeasible to search for a complete state space tree.

In order to improve the minimax algorithm search complexity, the author will explore alpha-beta pruning [2] on top of the minimax algorithm. This pruning algorithm adds two parameters for each tree node, α and β, where α means the best value can be achieved for the MAX player along the path to the root, and β means the best value can be achieved for the MIN player along the path to the root. Because the MIN player tries to make β as low as possible, at the MAX nodes level, if a node's child returns a value V which is larger than β of the node, then the node's other children can be cut off. Similarly, at the MIN nodes level, if a node's child returns a value V that is smaller than α, then its other children can be cut off. It is worth noting that the alpha-beta pruning technique does not affect final decisions. In the optimal situation where everything is ordered right, the time complexity of the minimax algorithm with alpha-beta pruning is $O(b^{d/2})$. Compared to the minimax algorithm, it can perform double the depth search within the same amount of time, or do half less work when searching the same depth of the state space tree.

The author will also implement a greedy approach that was mentioned in paper [3]. The algorithm works as follows: Assume it is the black's turn, then the black player checks whether its score is higher than the white player in the current state. If so, the black player will place the stone in a position that will maximize its advantage. Otherwise, the black will place the stone in a position that will maximize the white player's advantage. It is a simple approach because the player only looks ahead precisely one step and chooses to offend or defend based on the current state's scores.

In the future, the author will find more references about the Gomoku AI.

III. Evaluations

In this project, the author will simulate a two AI agents competition environment. AI agents can utilize the minimax algorithm, the minimax algorithm with alpha-beta pruning, or the greedy algorithm. The author is interested in comparing two AI agent teams: (i) the greedy AI vs. the minimax AI, (ii) the minimax AI vs. the minimax with alpha-beta pruning AI. In addition, the author will explore the effect of depth limitation on minimax with alpha-beta pruning AI. Two evaluation metrics will be adopted to evaluate their performances:

- Winning ratio: It is defined as the number of winning trades divided by the total number of trades.
- Average time per step: It is defined as the average time an AI agent spent to make a single decision.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[3] H. Liao, "New heuristic algorithm to improve the minimax for gomoku artificial intelligence," 2019.