Southwestern University

Evolution of Board Game Playing Agents

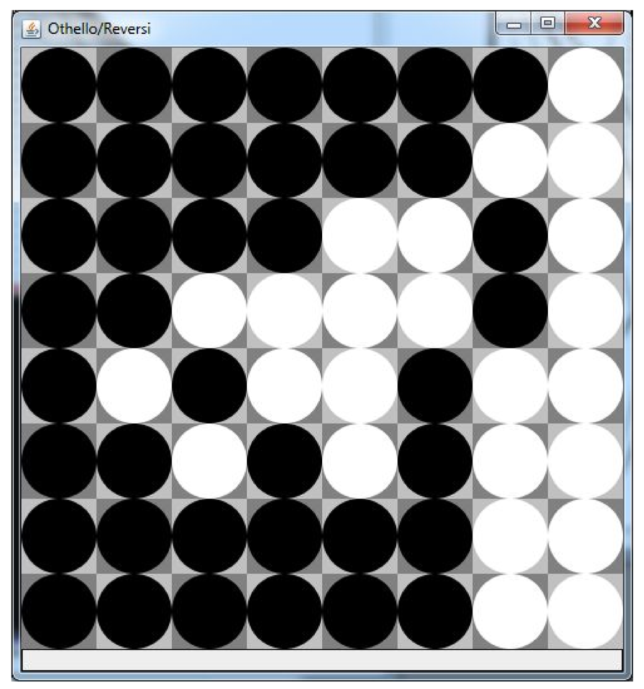
Sarah "Darwin" Johnson

Introduction

Board games are a common testbed for AI algorithms. Their simple design makes the game easy to understand and recreate. However, due to the number of possible moves, perfectly designing an agent to play any single board game and defeat a human player is a difficult and interesting challenge. One process that can create effective agents is using Neural Evolution of Augmented Topologies (NEAT [1]) to evaluate game states. This work uses NEAT to evolve state evaluators for an agent that plays the board game Othello against a static opponent.

Rules for Othello:

- Players take turns placing their color Chip on the Board; Black always goes first.
- Chips must be placed in a line that begins with a Chip of the Player's color and an unbroken line of Chips of the Opponent's color.
- The Chips that were between the placed Chip and the previous Chip are flipped over to the Player's color.
- If a Player cannot make a move, they must pass their turn.
- The game ends when neither Player can make a valid move.



Piece count:

Black (*Computer*): 40
White (*Human*): 24

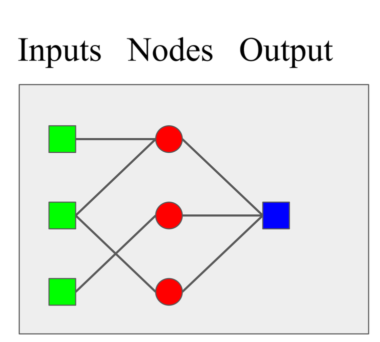
Minimax Search

All of the evolved agents and the static opponent use a Minimax Search algorithm when selecting a move. First, the agent considers each of the moves it can take from the current board state. Then, it calculates what moves the opponent could take from each resulting board state. This process continues until a specified depth, thus creating a game tree. At the bottom level of the tree, each resulting game state is evaluated. The board states at the final depth are given a score calculated either by an evolved neural network, or a static weighted-piece counter (WPC) [2]. Regardless, each player is assumed to pick the branch of the tree that leads to the best score.

White wants to maximize its score, and Black wants to minimize White's score, hence at each level of the tree either the maximum or minimum board state score is chosen. This process continues back up to the top of the tree, resulting in the best move chosen. The time taken to complete this process can be decreased by using Alpha-Beta Pruning to filter out certain game states without evaluating them.

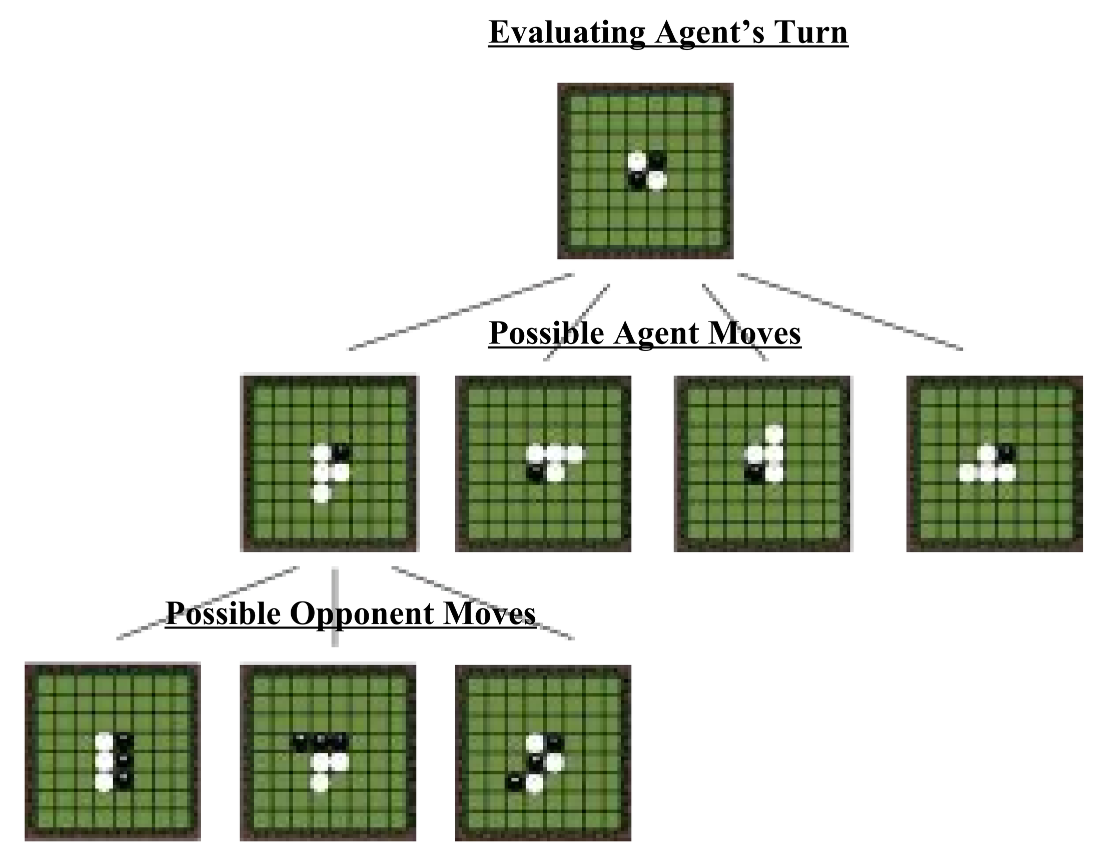
NEAT

The agents were evolved using a process called Neural Evolution of Augmented Topologies (NEAT), which evolves neural networks with arbitrary structure to approximate complex mathematical functions. These functions are then used to evaluate the board game states, creating unique behavior from the evolved agents. The structure of neural networks can be passed down to the next generation in their offspring, but with slight variation via crossover and mutation.



Experiments

Every agent and opponent had a 5% chance of selecting the second-best legal move rather than the intended best move, so the experiment was repeated 10 times to reduce the impact of sheer random chance. Each experiment used Single Population evolution against a static opponent. NEAT was used to evolve the evaluating networks [1]. The agents used Minimax tree-search with Alpha-Beta Pruning to evaluate the game tree.



Every agent played against a static opponent that didn't evolve and was consistent for every generation. The opponent used a Weighted Piece Counter (WPC) to evaluate the possible moves it could take. The WPC has values for every space on the board based on its location, and the values for every space are added to the board's overall score if there is a piece on it. Pieces representing the WPC opponent have a positive multiplier, and pieces representing the evolved agent have a negative multiplier [2].

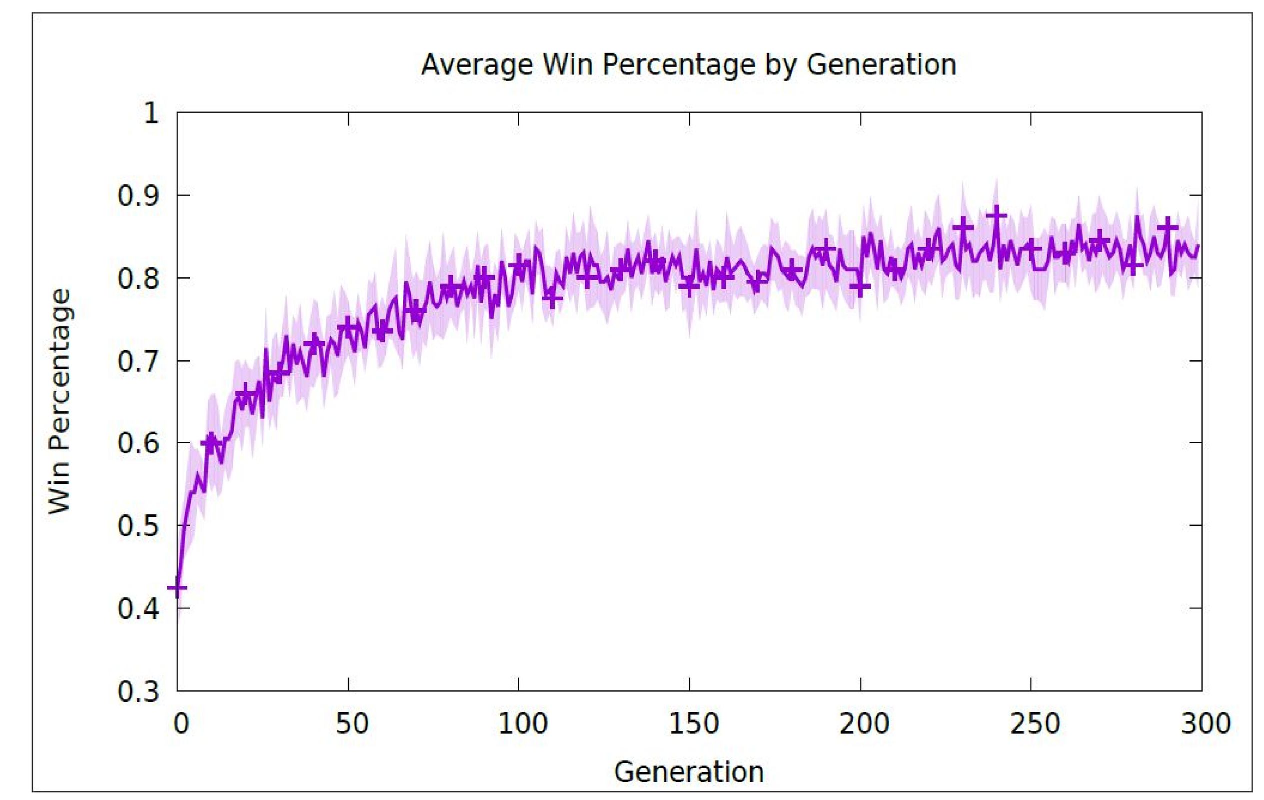
Each experiment had the following settings:

- 300 Generations were evolved
- 50 Parents per Generation
- 20 Trials against the WPC opponent; 10 as White +10 as Black
- 5% chance of choosing the second best move
- Alpha-Beta Pruning Minmax Search Agents
- Minimax Search Depth of 2
- Muliobjective Fitness Functions via NSGA-II [3]
  - Piece Differential
  - Win-Lose-Draw Score
  - Win Rate

The win percentage rate of each agent and the difference in the number of game pieces at the end of each game was recorded, and the results were averaged out for each evolved generation. In addition, each agent recieved a score based on the end result of a game; winning scored a 2, losing scored a -1. and a draw scored a 0. The scores were averaged out per agent, and the resulting scores were averaged out for each generation. The following chart shows the averaged win rate results across the 10 experiment runs.

Results/Discussion

The agents have an increase in average win percentage, ending piece differential, and win-lose-draw score for the first 100 generations of evolution. However, there is no significant improvement for the rest of the experiment; the agents from the 300th generation have nearly the same scores in every category. As such, there is no marked change in playstyle for the agents. These results signal that more tools need to be utilized when creating the agents in order to improve beyond the 100th generation.



Post-Evaluation Win Percentage of Final Champions Across 100 Evaluations by Trial #

Trial #	1	2	3	4	5	6	7	8	9	10	Avg
Win %	70.5	72.0	69.0	82.5	78.9	78.0	85.0	83.0	76.0	78.5	77.3

The win percentages of the final evolved agents against the static opponent across 100 evaluations average out to 77.3%. This means that the final result achieved during evolution is fairly robust, even though a higher win percentage would be better. Improved training scenarios are needed to evolve networks with even higher win rates.

References

[1] Kenneth O. Stanley and Risto Miikkulainen. Evolving Neural Networks Through Augmenting Topologies.
[2] Simon M. Lucas and Thomas P. Runarsson. Temporal Difference Learning Versus Co-Evolution for Acquiring Othello Position Evaluation.
[3] Kalyanmoy Deb, Amrit Pratap, and Sameer Agarwal. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.