SOUTHWESTERN UNIVERSITY

Using Neuroevolution on Checkers Players

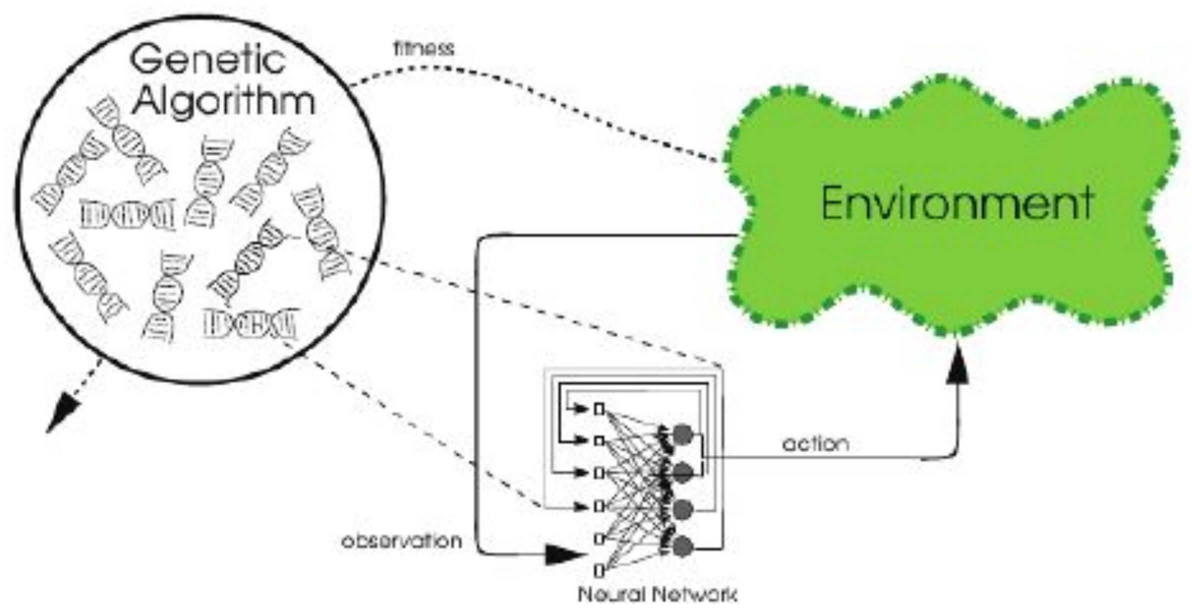
Ryan Wheeler (in collaboration with Nicholai Benko and Guy Greathouse)

Introduction

Checkers has been the focus of extensive artificial intelligence research. We use MM-NEAT, an algorithm for evolving artificial neural networks, to evolve an agent capable of playing at a high level. Expert knowledge about the game of Checkers was used to design features to help accelerate the learning process.

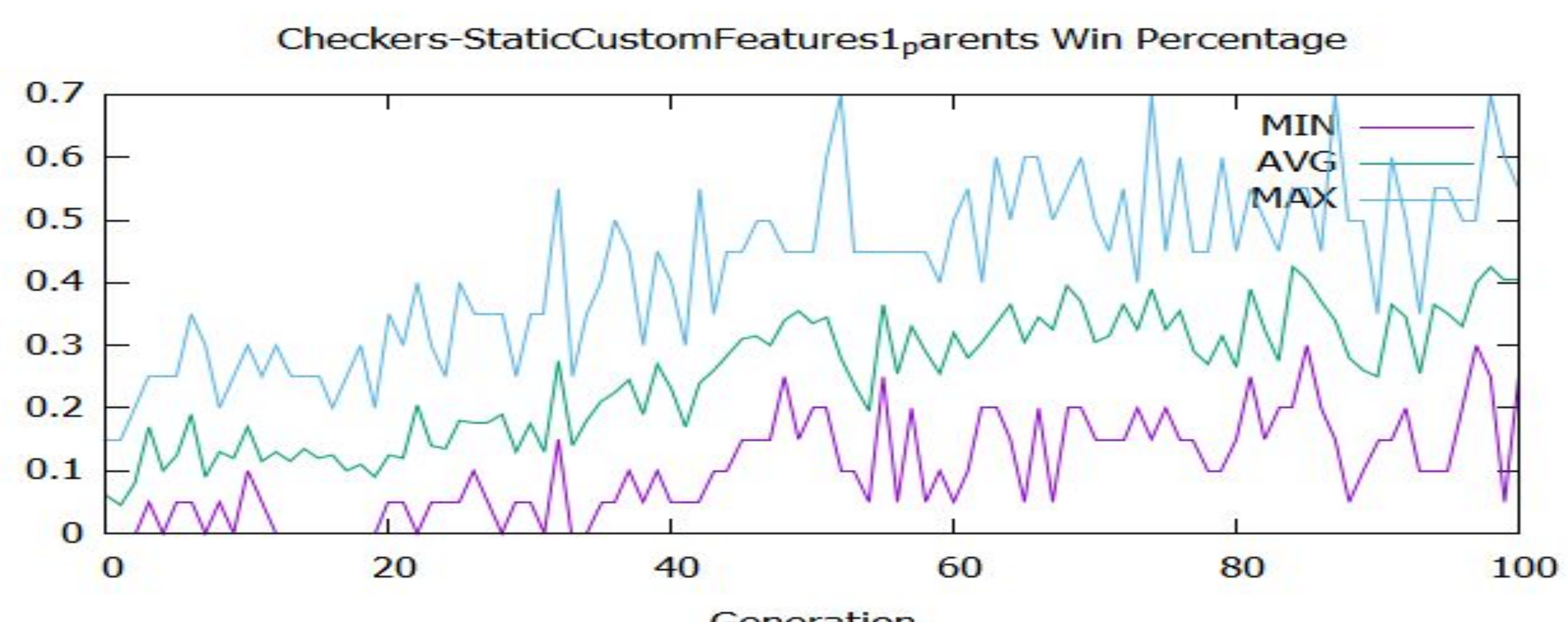
MM-NEAT

Multiobjective Modular Neuro-Evolution of Augmenting Topologies (MM-NEAT) is a software framework that builds on the basic ideas of Neuro-Evolution of Augmenting Topologies (NEAT) [2]. NEAT networks start small and simple and grow more complex through structural mutations. The NEAT algorithm models evolution through natural selection to favor successful networks and random mutation to create new networks from successful networks. This is an attempt to find the balance between optimizing the fitness function of evolved solutions and generating new diverse options that could potentially solve the problem more effectively.

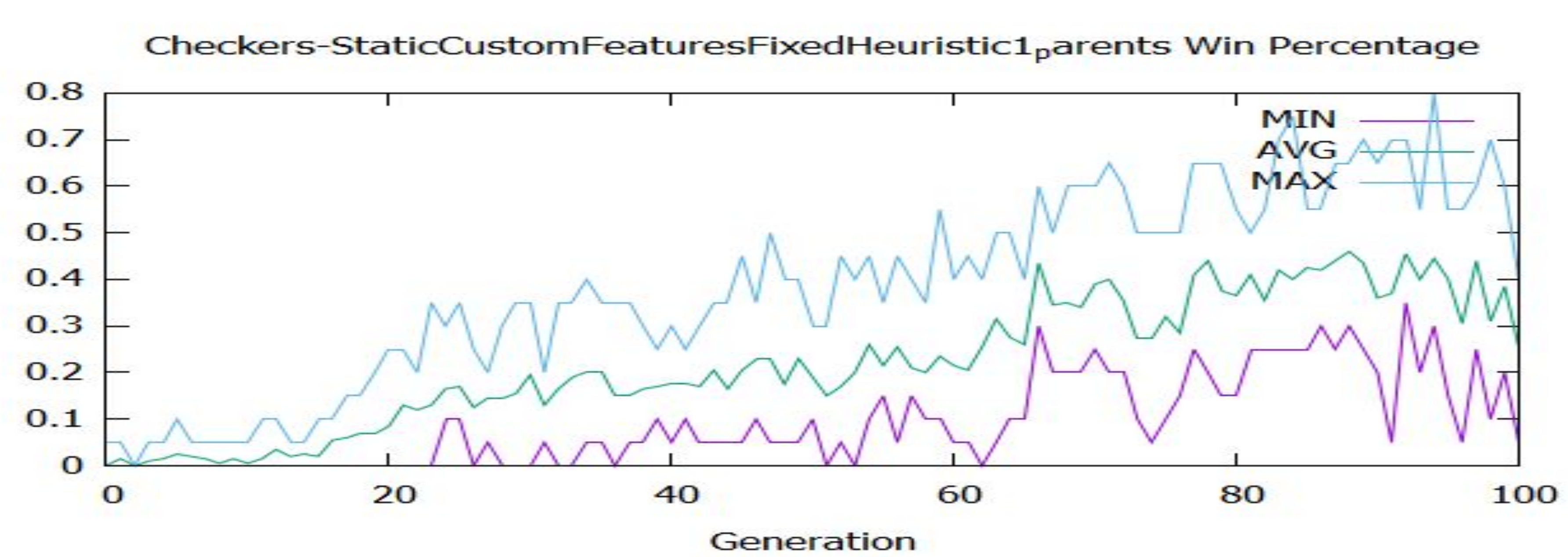


Experimental Procedure

- The three factors we were changing were: the feature set that our player used to evolve, the opponent type (random or heuristic), and the heuristic that the opponent used.
- We ran a minimax agent with a the raw board feature extractor, playing against a random opponent to use as a baseline to which we compared all future tests
- Nine new features were added to the raw board game inputs: count of checks and kings for each player, individual piece differentials for both types of pieces, number of pieces that could capture an opponent's check, number of pieces in danger of being captured, number of pieces on the edge of the board [1]
- We first evolved the raw board state input against the sophisticated heuristic. Subsequently, we evolved our custom agent against the sophisticated heuristic.



Win Percentage of Raw Features vs Random Player Over Time



Win Percentage of Advanced Features vs Basic Heuristic Over Time

Results

- We achieved close to 70% win rate against the random opponent and 80% against the basic heuristic agent for 100 generations.
- Results in terms of overall win percentage against the advanced heuristic were not ideal
- Our agent showed an improvement in win percentage and fitness against the sophisticated heuristic compared to the simple feature set of only raw board input.
- Our features did have success in increasing agent performance against a strong opponent

Conclusion

This project demonstrated the effectiveness of our custom feature set in comparison to the raw board state using MM-NEAT. Our project showed success in using expert knowledge features to evolve a checkers player. While demonstrating success, our results can be improved. Increasing generation size and adjusting trial and population size are small improvements worth trying in the future.

References

[1] Jacek Mańdziuk, Magdalena Kusiak, and Karol Walędzik. 2007. Evolutionary-based heuristic generators for checkers and give-away checkers. Blackwell Publishing Ltd. (August 2007), 23 pages. DOI: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1468-0394.2007.00429.x

[2] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks Through Augmenting Topologies." Evolutionary Computation, 10(2):99-127, 2002. URL: http://nn.cs.utexas.edu/keyword?stanley:ec02