Evolution of a Competitive Bot Bowl Agent Using Genetic Algorithms

Benjamin Ledoux, Gary Parker, Jim O'Connor

Abstract

We discuss the methods by which we evolved a rules-based expert system for a competitive agent to submit to the Bot Bowl competition using Genetic Algorithms (GAs). The expert system is evolved by identifying variables that can be refined through the GA, and measuring the success of each variation by its performance against a baseline opponent. A variation is derived from a string of bits, or "chromosome," that defines the rule set used by the agent. One of the unique challenges of using a GA in the Bot Bowl environment is the low- scoring nature of the game. Rather than using the score as the only measurement of success for the fitness calculation, we propose including the ball's progression towards the opposing end zone to quantify the performance of a particular chromosome.

Environment

Bot Bowl is an annual artificial intelligence (AI) competition hosted by the Institute of Electrical and Electronics Engineers Conference on Games. The competition is based on the game Blood Bowl: a "fully observable, stochastic, turn-based, modern style game with a grid-based board" based on the real-life sport of American Football (Justesen et al. 2019, 1). The Bot Bowl competition is a round-robin tournament between participating "bots" competing one-on-one for 10 games to determine which bot can play Blood Bowl the best, which is determined by their wins, draws, and losses against each other. A baseline bot is included in the tournament to serve as a reference point. This role is fulfilled by a "Random Bot" that selects

actions at random. The competition is standardized so that bots are required to use human teams of 11 players. There are two prizes awarded: (1) for the most successful bot overall, and (2) for the "most interesting bot that uses machine learning or search-based techniques" (Justesen 2024). In recent years, the most successful agents have been fully or partially rules-based. A rules-based, or "scripted" bot, is broadly defined as an agent that follows a static set of rules to dictate the agent's actions, those rules being defined prior to the start of the match. Our goal for this work is to design, implement, and submit a competitive agent using a rules-based expert system that has been optimized by evolutions through a GA to fine-tune variables used in decision-making.

Related Works

In 2019, alongside the first edition of the Bot Bowl, the creators of the competition published a paper on the Blood Bowl environment's potential as an AI research platform. Given the stochastic nature of the game, they proposed Blood Bowl as a "new challenge for Artificial Intelligence" (Justesen 2019, 1). Nicola Pezzotti's 2021 paper on their Mimic Bot submission details the combination of reinforcement learning and imitation to create the winning agent of the overall prize in Bot Bowl III. The paper highlights improved methods of training in reinforcement learning environments and improving sample efficiency (Pezzotti 2021). Dref Sante maintains a blog detailing updates on their rules-based system, which won both Bot Bowl IV and Bot Bowl V. The most recent update to the agent was in February 2024 with the release of version 0.7 (Sante 2024). Included in the Bot Bowl source code is the OpenAI Gym toolkit for reinforcement learning. Gym includes a general-use interface for reinforcement learning, which increases the speed of simulations using the toolkit, including Blood Bowl games. This speed of

simulation makes Gym ideal for developing agents that require competing in multiple games in order to train and improve (Brockman et al. 2016, 1).

Genetic Algorithms

A genetic algorithm, or GA, is an algorithm that is "designed to simulate [evolution]" (Carr 2014, 2). It begins with a randomized population of chromosomes, which are often implemented as bit strings, that represent different solutions to a problem. The following three steps are then repeated until a suitable chromosome is found: (1) fitness calculation, (2) parent selection and crossover, and (3) mutation. Fitness calculation is a measurement of the success of a particular chromosome. Similarly to its biological counterpart, this kind of chromosome with a high fitness value can be considered to have advantageous traits compared to others in its population, which is used to artificially increase that chromosome's influence on the following generation. Since each chromosome in our GA acts as a controller for the agent through the rules-based system, the fitness calculation is based on the performance of the agent in a game of Blood Bowl using the behaviors defined by the chromosome.

The chromosomes are then used to create a new generation, which is supplemented by the process of elitism. Elitism in GAs is the retention of the best-performing chromosomes across generations. Once the chromosomes with the highest fitness scores are copied over to the next generation, the rest of the population is filled by selecting parent chromosomes and combining them to make child chromosomes. The selection method for parent chromosomes in our GA uses a tournament style: we select some number of chromosomes at random, choose whichever chromosome has the best fitness out of that group as the first parent, and repeat the process for the second parent. The parent chromosomes are combined using a random crossover point in the chromosome. The child chromosome consists of all bits from the first parent up until

the crossover point, and the remaining bits from the second parent starting at the crossover point. This process of selection and crossover is repeated until the size of the population matches the previous generation.

Finally, the population is "mutated." Mutation is the process of randomly flipping bits in order to maintain variety in the population. Over time, the population is expected to begin to look similar as better-performing chromosomes contribute to more child chromosomes, so mutation is used to keep populations diverse. The process then begins all over again with the new generation of child chromosomes and repeats until a suitable chromosome is found. Our final chromosome will then be deconstructed, and its behaviors will be implemented as the agent's static rules-based system.

Challenges

The first major challenge in applying a GA to a Blood Bowl agent was developing the fitness function. The traditional measurement of performance in a sport or related game is how many points are scored, making it an ideal metric for calculating the fitness of a chromosome-based controller. In Blood Bowl, a single point is awarded whenever a player in possession of the ball enters the opposing team's endzone. However, Blood Bowl is often a low-scoring game when played by experienced people, with many games between experienced players ending in draws. This creates a difficult environment for a GA to evolve chromosomes, where scoring is rare enough to hinder chromosome evolution. In order to counteract the rarity of scoring in the fitness calculation, it was deemed necessary to incorporate a more common measurement of success akin to scoring. The solution chosen was to incorporate ball progression as the supporting measurement of fitness. As the ball is moved towards the opponent's end zone, the agent is more likely to score, so it would prioritize moving the ball into a scoring position.

While the final score is still highly valued in the fitness calculation, including ball progression allows for a constant metric that directly correlates to success.

The second major challenge to present itself was the accuracy of the fitness calculation. A player's success in Blood Bowl is largely dictated by the frequent dice rolls used to determine the outcome of many actions. Given the stochastic nature of Blood Bowl, chromosome fitness values could be affected by the luck of dice rolls in a game. If a strong chromosome happened to control the agent during a particularly unlucky game, that strong candidate would have less influence on the next generation, weakening the gene pool. Similarly, if a weak chromosome happened to control the agent during a particularly lucky game, that weak chromosome would affect the next generation negatively and possibly take the place of a stronger chromosome during the elitism step. To counteract this inherent randomness, we increased the number of games each chromosome controlled the agent for. By averaging the metrics used for the fitness calculation, the chromosomes were better represented by those metrics. The more games a chromosome controlled the agent for, the less likely that luck would heavily sway the outcome.

Method

In order to evolve the rules-based expert system, we first had to identify and extract variables to evolve through the GA. The first set of variables identified were binary choices on whether or not to use one of the "reroll dice" to attempt to gain a better outcome in different scenarios like blocking, catching, passing, and more. In Blood Bowl, certain actions can result in a "turnover" if they fail. A turnover results in the immediate termination of a player's turn, which can greatly affect the outcome of the game. Despite the importance of using reroll dice wisely, exposing that decision to the GA had very little effect on the performance of the agent over time, as a player is limited to only three reroll dice in each half of a game. The next set of variables

identified were “chance variables,” where certain courses of action only took place if the calculated chance of success met a certain threshold. A “safe block” would only be attempted if the chance of success was above 94%. By exposing these chance variables to the GA, the agent can determine its own definitions of “safe” versus “risky” thresholds through evolution.

At this point our agent was still not exceeding the performance of the original rules-based expert system, so the next variable identified was the ordering of actions. Originally, actions were ordered from least likely to result in a turnover to most likely. Some of these actions were reliant on the previously mentioned chance variables but still maintained a constant order of operations. Even after evolving its own order of operations, the GA could only produce an agent that marginally exceeded the rules-based expert system.

Results

While the GA did improve our agent’s performance against the Random Bot when compared to the original rules-based expert system, the difference was not significant enough to estimate our agent would place high in the competition in the next Bot Bowl competition. Looking at the results of previous years, we estimate our system would place in the middle of the table. We are confident it will beat the Random Bot in all ten games and can perform similarly against other submissions that struggled against the Random Bot.

The poor results are partially due to a lack of time. Simulation of games without any rendering took upwards of 12 seconds on average, so evolving 100 chromosomes for only 100 generations would take upwards of over 33 hours. This assumes each chromosome only played one game, meaning the averaging of multiple fitness scores per chromosome across 5 games would increase that time to almost a full week to complete. This time to complete a single game

is not unprecedented, as Dref Sante mentions in their most recent update on their agent that simulations of games take about 11 seconds to complete (Sante 2024).

We also found that the best-performing chromosome of each population was increasing too quickly at the beginning. It never took more than five generations for the best-performing chromosome to exceed the performance of the original rules-based expert system against the Random Bot.

Future Work

There are two primary means of improving our system's performance. The first is to deconstruct more source code and add our own rules. While the agent itself has been broken down extensively, some decisions utilize external code, such as path finding, that can be replaced or modified and included alongside new rules in the evolution through the GA. This would likely slow down the speed at which the chromosomes are evolved, giving us a more natural fitness curve.

The second method of improving our agent's performance is adapting the GA into neuroevolution. Neuroevolution is the evolution of Artificial Neural Networks through a GA similar to the one used in our research. A Neural Network (NN) simulates how a human brain works using connected nodes that predict the outcome of an inputted scenario. By adapting to neuroevolution and evolving an Artificial NN for the agent, the agent can be submitted with an improved ruleset that analyzes the state of the game board and determines actions based on the prediction.

References

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. "OpenAI Gym." arXiv preprint arXiv:1606.01540.

Carr, Jenna. 2014. “An Introduction to Genetic Algorithms.”

Justesen, Niels, Lasse Møller Uth, Christopher Jakobsen, Peter David Moore, Julian Togelius, and Sebastian Risi. 2019. "Blood bowl: A New Board Game Challenge and Competition for AI." In 2019 IEEE conference on Games (CoG), pp. 1-8. IEEE.

Justesen, Niels. 2023. “A Blood Bowl AI framework.” GitHub. Accessed February 26, 2024. https://github.com/njustesen/botbowl.

Justesen, Niels. 2024. “Welcome to Bot Bowl | Bot Bowl.” GitHub Pages. Accessed February 26, 2024. https://njustesen.github.io/botbowl/.

Pezzotti, Nicola. "MimicBot: Combining Imitation and Reinforcement Learning to win in Bot Bowl." arXiv preprint arXiv:2108.09478 (2021).

Sante, Dref. 2024. “Blood Bowl AI Implementation.” Blood Bowl AI Implementation. https://drefsante.blogspot.com/.