Hybrid Fighting Game AI Using a Genetic Algorithm and Monte Carlo Tree Search

Man-Je Kim
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
jaykim0104@gist.ac.kr

Chang Wook Ahn*
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
cwan@gist.ac.kr

ABSTRACT

Real-time video game problems are very challenging because of short response times and numerous state space issues. As global companies and research institutes such as Google, Facebook, Intel and Carnegie Mellon university continue to develop Artificial Intelligence(AI) and participated in various Game AI competitions, many AI developers are implementing algorithms, such as Monte Carlo Tree Search (MCTS) and Deep Reinforcement Learning (DRL), to solve problems. However, these algorithms also have a number of limitations, including the inability to effectively calculate the number of possible cases present in the games and the large amount of time required for computation. We combine the genetic algorithm, which is very effective in finding general solutions to solve the constraints of AI, and the MCTS, which dominates the recent Fighting Game AI Competition, into a hierarchical structure. We propose to call it Hybrid Fighting Game AI.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**; **Evolutionary algorithms**; **Bio-inspired optimization**;

KEYWORDS

Artificial Intelligence, Evolutionary Computing and Genetic Algorithms, Global Optimization

ACM Reference Format:
Man-Je Kim and Chang Wook Ahn. 2018. Hybrid Fighting Game AI Using a Genetic Algorithm and Monte Carlo Tree Search. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205695

1 INTRODUCTION

The IEEE Conference on Computational Intelligence in Games (CIG) is hosting a variety of video games for Artificial Intelligence (AI) competitions. Among them is a fighting game in which two AIs fight in one space with a limited response time of action less than or equal to 16.6ms (1/60 sec), which is very short compared to that of turn-based games.

As a result, the AIs have evolved to find a large number of cases more efficiently than other fighting game AIs. Among these methods, Monte Carlo Tree Search (MCTS) has shown the best results.[3] The fighting game AIs based on MCTS won the 2017 IEEE CIG Fighting Game AI Competition after their release in 2016. In order to find a good solution within a short response time, they decide the best action by randomly selecting and simulating 5 possible actions in current states with their opponent.

In addition, MCTS-based AIs use a mixture of certain rules for distances between characters and an action prediction table that analyzes opponent AI behavior from the past in order to select the five actions mentioned above.[2] However, most of these AI action choices are heavily dependent on human domain knowledge. Although the AI has achieved excellent results, the developers have failed to explain why it performs well, and ultimately, which direction it tends to choose. Such a development approach is likely to have a negative impact on the development of artificial intelligence technologies for the games of this genre. So, we are trying to solve the aforementioned problems with Hybrid Fighting Game AI using genetic algorithm(GA)[1] and MCTS[3], which would demonstrate the possibility that GA can be used in the real time video game artificial intelligence development.

2 GENETIC ALGORITHM FOR FIGHTING GAME

Each playable character in the fighting game introduced here has 56 possible actions in total. A player can actively select 40 of them (air actions: 15, ground actions: 25). In addition, since a player needs to predict the actions of an opponent due to the nature of real-time video games, there are very many cases (40x40=1600) out of which to make the best choice in a certain situation. states cannot overlap other states, so there are too many possible combinations to simulate within 16ms, even if one cautiously chooses each one.

Consequently, we used a genetic algorithm to find and select five viable actions, and we developed artificial intelligence by constructing a hierarchical structure to select the best action based on MCTS. We have conducted the following experiment to prove that this hierarchy can be applied well to the fighting game.

Table 1 shows the settings we used to apply the GA to the fighting game. The chromosome size is set to 5 because there are 5 actions that can be simulated in the MCTS. Duplication of the gene in each chromosome is not allowed because it simulates all 5 orders of action. The gene was randomly selected so that it did not overlap within the chromosome. In case of the crossover method, since duplicated genes are not allowed in the chromosome, the two

*The corresponding author

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205695

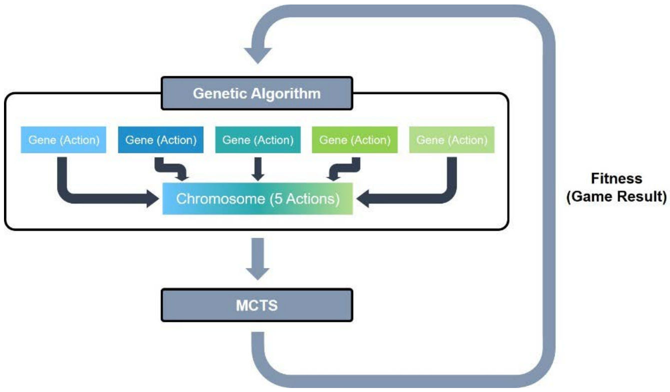

Figure 1: hierarchical structure of GA and MCTS

Table 1: Default Genetic Algorithm Setting

Element	Contents
Gene	Action
Chromosome	Bundle of non-duplicated actions
Fitness	Gap of HP (My HP-Opponent HP)
Selection Method	Tournament Selection (K=0.95)
Crossover Method	Non-duplicated Two-point Crossover
Population Size	20
Chromosome Size	5
Mutation Probability	0.005

leftmost actions are interchanged except for the duplicated ones. If there are less than 2 crossable actions, then only the leftmost one will cross. We set the difference in fitness per game as the gap of Hit Points(HP) because it is very unstable when determining fitness in real time. For each round, an opponent and our AI act concurrently, so a very wide variety of situations is unfolding, causing the AI to comply with the situation even if it has better choices of action. This is because periodically making decisions every minute usually results in bad fitness and convergence. To solve this problem, we decided to use the fitness as the difference in HP between the opponent and the AI to obtain a relatively stable fitness by evaluating various situations within the game.

3 RESULTS

To evaluate the performance of the AI, we measured the output of each generation against Basic MCTS AI. The result is shown in Figure 2. We iterated 10 generations in this experiment, but when the generation turned between the 8th and 9th cycles, out of 10 crossover cycles, the applied chromosome appeared to be the same for 5 times, a single gene appeared differently for 4 times (thus not replaced), and finally only 1 crossover occurred. As a result, the chromosomes of the 8th and 9th generations were hardly different from each other, so we show only generation 8 in Figure 2. In fact, the change in the values was not large. In addition, our AI won 99

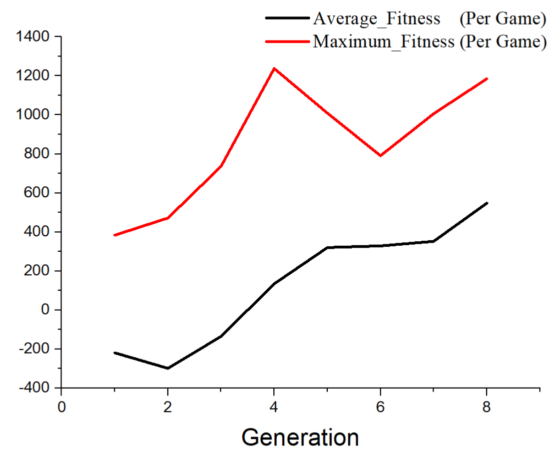

Figure 2: Performance change of Hybrid Fighting Game AI against Basic MCTS AI

games out of 100 games against the Basic MCTS AI, utilizing the chromosome with the maximum fitness in generation 8.

4 CONCLUSIONS

We have developed Hybrid Fighting Game AI in a real-time fighting game by combining the GA and the MCTS. While the GA alone is very good at finding a good solution, it is not very well suited for real-time games. This is because it is difficult to solve the real-time video game problems with a structure that can utilize the GA in real time due to the amount of computation too large for the short response time of real-time games. However, we have proposed a method to solve the problem of short response time by applying the GA and MCTS hierarchically. In addition, it has the advantage that it can utilize algorithms other than the MCTS that can simulate in real time. Since our structure does not utilize any domain knowledge, it can be applied to other games as well. As we have discussed, Combining the strengths of two or more algorithms into a structure can solve problems more efficiently.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2015R1D1A1A02062017).

REFERENCES

[1] John H. Holland. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* MIT Press.
[2] Kim Man-Je and Kim Kyung-Joong. 2017. Opponent modeling based on action table for MCTS-based fighting game AI. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 178–180.
[3] Yoshida Shubu, Ishihara Makoto, Miyazaki Taichi, Nakagawa Yuto, Harada Tomohiro, and Thawonmas Ruck. 2016. Application of Monte-Carlo tree search in a fighting game AI. In *2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE, 1–2.