2C4-IOS-3c-8

An Efficient Framework for Winning Prediction in Real-Time Strategy Game Competitions

Chiu-Jung Hsu
Department of Computer Science and Information Engineering
National Chung Cheng University, Taiwan
shsucs@yahoo.com.tw

Shao-Shin Hung*
Department of Computer Science and Information Engineering
WuFeng University, Taiwan
hss@cs.ccu.edu.tw

Jyh-Jong Tsay
Department of Computer Science and Information Engineering
National Chung Cheng University, Taiwan
tsay@cs.ccu.edu.tw

In recent years, real-time strategy (RTS) games have gained attention in the AI research community for their multitude of challenging and relevant real-time decision problems that have to be solved in order to win against human experts or to collaborate effectively with other players in team games. However, a big challenge for creating human-level game AI is the different traits of races of opponents and their locations of enemy units are partially observable. To overcome this limitation, we explore evolutionary-based approach for estimating the location of enemy units that have been encountered. In this paper, we propose an efficient framework to predict the winning ratio between the different races used in the real-time strategy game. We represent state estimation as an optimization problem, and automatically learn parameters for the evolutionary-based model by learning a corpus of expert StarCraft replays. The evolutionary-based model tracks opponent units and provides conditions for activating tactical behaviors in our StarCraft bot. Our results show that incorporating a learned evolutionary-based model improves the performance of EISBot by 60% over baseline approaches.

1. Introduction

Computer games are situated in a virtual world, involve a variety of player skills and decision making processes, and provide a fun immersive experience. On the other side, a Real-time Strategy Game (RTS) [Jaap et.al. 2005] [Churchill 2011] is a game in which the players use resource gathering, base building, technological development and unit control in order to defeat its opponent(s), typically in some kind of war setting. The game is not turn-based in contrast to board games such as Risk and Diplomacy. Instead, all decisions by all players have to be made in real-time. Usually, it encompasses a subset of academic AI techniques that implement adhoc solutions in three groups [Jaap et.al. 2005]: (1) Movement mechanisms, providing the decision process to control NPC's motion, e.g. optimised real-time versions of A_ algorithms; (2) Behavior control used to control NPCs' actions; (3) Strategy techniques used to co-ordinate groups of NPCs. In this paper, we propose one evolutionary-based approach for estimating the location of enemy units that have been encountered to predict the winning ratio between the different races used in the real-time strategy game (i.e, in StarCraft) [Churchill 2011].

2. Related Works

In general, opponent modeling can be seen as a classification problem, where data that is collected during the game is classified as one of the available opponent models [Hiong 2011]. One limiting condition is the fact that in RTS games, these classifications have to be performed in real-time. This limits the amount of available computing resources, which is why only computationally inexpensive techniques are suitable for opponent modeling in RTS games. Commercial game AI provides an excellent baseline for agent performance, because it must operate within a complex environment, as opposed to an abstraction of a

Contact: Shao-Shin Hung; Department of Computer Science and Information Engineering of WuFeng University; Chiayi Taiwan; hss@cs.ccu.edu.tw

game. However, the goal of commercial game AI is to provide the player with an engaging experience, as opposed to playing at the same granularity as a player. Both deliberative and reactive planning approaches have been applied to games.

3. Our Evolutionary-based Approach

Here, we will introduce our evolutionary-based approach. First, the neural network (NN) [Churchill 2011] will be the basic component and algorithm. Based on the principles of evolutionary approach, the Q-value, CrossOver value, number of wining times, winning probability, and History table for recording the each state for results of NN execution are all concerned. At the initial point, the three different races, named Protoss, Terran, and Zerg. We focus mainly on the wining probability value for Zerg and two other races. The interval time of evolution is set as five. This means that every five generation of evolution will check their Q-value is higher or lower. If higher, the new cross-over generation will replace the old one in the History table (i.e. Zerg); otherwise, the new generation will not replace the current one. All experimental values are recorded in the History table. If needed, we can search the table and modified the corresponding value if possible. Besides, all opponent races are selected randomly and same to the selection of its corresponding NN.

4. Discussions

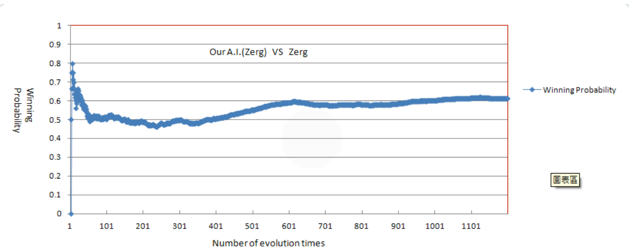

Figure1. The winning probability is measured our game AI and between non-game AI mechanisms for Zerg and two other races competitions.

We will introduce our experimental environments. CPU is i5 with 2300, RAM is 4G, hard disk is 500G, and window system is Win 7 professional version. As for the input data, we adopt Broodwar (1.16.1) with API 3.74. The average wining probability rate is above 0.612. If we choose the other two races for evolution basis to compete with Zerg, the winning probability will drop and is less than 0.2. Apparently, this result still has some room for improvement.

5. Conclusions and Future Works

In this paper, we present state estimation as an optimization problem, and automatically learn parameters for the evolutionary-based model by learning a corpus of expert StarCraft replays. The evolutionary-based model tracks opponent units and provides conditions for activating tactical behaviors in our StarCraft bot. Our results show that incorporating a learned evolutionary-based model improves the performance of EISBot by 60% over baseline approaches.

In the near future, we will improve the winning probability rate by present more efficient and effective algorithm to gain better results.

References

[Jaap et.al. 2005] Jaap van den Herik, Jeroen Donkers, and Pieter Spronck. Opponent modelling and commercial games. Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games, pp. 15-25, 2005.

[Michael 2012] Michael Buro and Davis Churchill, Real-time strategy game competitions. AI Magazine, Association for the Advancement of Artificial Intelligence, pp. 106-108.

[Weber 2010] Ben G. Weber. Peter Mawhorter, Michael Matesa, Arnav Jhala. Reactive planning idioms for multi-scale game AI, IEEE Conference on Computational Intelligence and Games (CIG 2010).

[Hiong 2011] Chin Hiong Tan, Kay Chen Tan, and Arthur Tay. Dynamic game difficulty scaling using adaptive behavior-based AI, IEEE Transactions on Computational Intelligence and AI in Games, Vol.3, No. 4, pp. 289-301, December 211.

[Churchill 2011] Churchill, D., and Buro, M. Build Order Optimization in Star on Artificial Intelligence and Interactive Digital Entertainment, pp. 14–19. Menlo Park, CA: AAAI Press.