Apollo2D Team Description Paper

Yongbo Li[1], Chang Lin[1], Siyuan Yu[1], Yifan Chen[1], Xinyi Jie[1], Minghui Jiang[1]

[1] Nanjing University of Posts and Telecommunications
apollo2d@outlook.com

Abstract. This team description paper mainly explains the work of Apollo2D recently. We emphasize on this document about characteristics of our team which we devoted ourselves to, including passing cooperation and the related projects called HFO Trainer. They are steady points towards the promotion of the team's performance.

Keywords: RoboCup, 2D Simulation, A*, Half Field Offense.

1 Introduction

1.1 Overview

Apollo2D is a part of Apollo which serves as a robotics club and a competition team affiliated to Nanjing University of Posts and Telecommunications (NJUPT for short).

1.2 Team History

Apollo2D team went through inconsistent time as below.
Founding history. Apollo was founded in 2002 as a competition team under the leadership of Professor Zhiyong Zhang, when the team mainly focused on 2D simulation. The team made up a fundamental RoboCup environment and got achievements, such as the third place in RoboCup U.S. Open 2005.
Midterm history. The organizational form of Apollo changed into a robotics club in 2010, when the main emphasis of Apollo switched to RCSS 3D and Rescue, which means 2D got less attention or resources. What's worse, Apollo2D lacked communication between generations, as a result of documents and data loss, following with decreasing efficiency. The worst year is 2018, no freshmen participated, causing a one-year-long break of our team.
Recently. 2 postgraduate students took on research about improvement of Apollo2D agents since 2016. Undergraduate students in 2019 starts to revive the group, including increasing people. Apollo2D won the second place in RoboCup China Open 2021.

1.3 Team Code

We based our code on agent2d 3.1.0[1] until this year. Now Apollo2D uses up-to-date HELIOS base including librcsc rc2021. HELIOS base[2] is a great bridge connecting

RoboCup Soccer Simulation (RCSS for short) Server and the detail implement of a team. Also, it guarantee stability and potential of a team.

There are obvious trails that shows a small part of our codes originated from Yushan and Tianjin Normal University (TJNU for short), but in old version which cannot be dated. We mainly refer to build system and train of thought from Yushan and behavior like pass and block from TJNU.

Over the past few years, the focus has been on two aspects of formation design and pass cooperation, proposing a formation design method based on a pass cooperation strategy based on A* search.

2 Pass cooperation strategy based on A* search

2.1 Tree structure model and A* search deployment

Tree structure model. A chain of pass in a small range is treated as a tree model[3] in Apollo2D, in which a pass action serves as a node and it is paired with a state to predict[4].

A* search. After simplify the soccer field to trees, Apollo2D runs A* on these pass tree. Since A* uses priority queue to select paths that minimizes[5][6]

$$f(n) = g(n) + h(n) \tag{1}$$

In RoboCup Soccer Simulation environment, we define $g(n)$ is 100 * possibility of being intercepted, $h(n)$ is the distance between positions of ball and goal, just as shown in **Fig. 1**, where the values on the nodes are $h(n)$, the values on the arrays are $g(n)$.

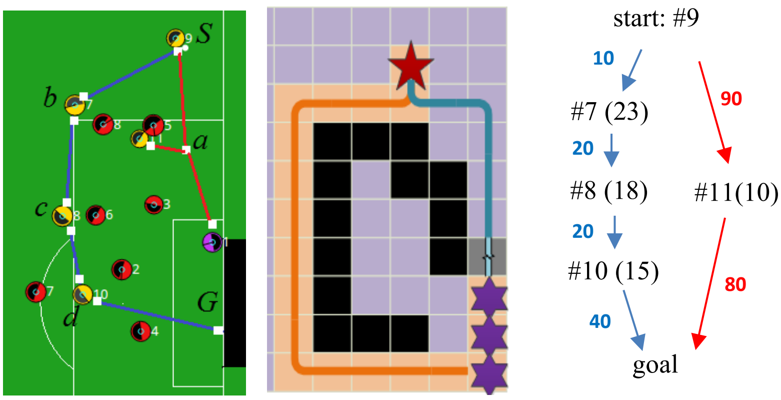

Fig. 1. Transformation to A* algorithm model on a passing problem

Weight involvement. In order to enhance flexibility of our team when facing different situations, we need to adjust $g(n)$ and $h(n)$, so **Equation (1)** was modified to[7][8]:

$$\begin{cases} f(n) = a * g(n) + b * h(n) + w \\ a + b = 1 \end{cases} \tag{2}$$

Amount balance between weights a, b and w is decided according to the strategy. We also divided the soccer field into 8 blocks as indicators on the decision of the weights[9][10].

2.2 Results of the improvement

After we try to get the best value of a, just as shown in **Fig. 2**, the value around 0.5 seems like the answer. Judgement of the game stage and carrying out more flexible data is also a solution, which is the practical application.

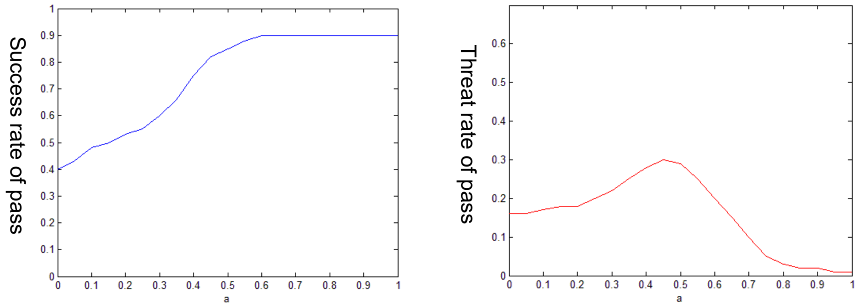

Fig. 2. Two rates about pass under influences of value a

The through balls with attack threat, which successfully breakthrough the defensiv e players only by passing and make shoot chances in the end, threat rate of pass is the ratio of through ball to the total number of passes. **Table 1** shows data analysis of Apollo2D vs Alice2021 in 18 games. The way to analyze is Loganalyzer3[1].

Table 1. Data Comparison

	Apollo2D	Alice2021
Total scores	23	18
Threat rate of pass	**2.774%**	**2.354%**
Domination cycles (average)	2179.72	1910.72

3 HFO Trainer

3.1 Overview of Half Field Offense

Half Field Offense[11] environment is a branch of 2D simulation which pays attention to the offense-defense situations within the half field. It's established around a rcssserver (version 15.2) addition to a HFO referee, with involvement of Python

[1] loganalyzer3 https://github.com/opusymcomp/loganalyzer3

scripts and other tools. Users can connect to it with their own agents and practice them.

3.2 Shortcomings of its environment

Although it's a complete and high-available system, we noticed some shortcomings. The HFO referee inside HFO system is the core element of the main function, but due to the referee class is edited inside the source code, so here it brings some problems of inflexibility.

Problems on version update. When rcssserver upgrades to version 16, a lot of new characteristics cannot be applied on old HFO environment. It takes effort to make HFO part fit in new server.

Conflict with official rcssserver. Since its rcssserver has been modified and add some new variable to configuration files, but it's still read config from default directory which places the default config files generated by official rcssserver.

3.3 Migration to a RCSS Server Trainer

Taking shortcomings above into consideration, we would like a better environment where agents are trained. Trainer in RCSS Server has access to control variables on the field, just like referee. It proves the feasibility of migration.

In detail, we imitate sample trainer, integrate referee contents in to two organized classes, which is injected into main function of HFO trainer. To make it work well, necessary parse command is added.

We have finished approximately 80% of the work in the HFO system.

3.4 Features or improvements of HFO Trainer

File structure. HFO Trainer is an executable file parallel with player, sample trainer and coach in a team. Codes inside this directory are used to set HFO training environment.

Independence and flexibility. In theory, all different sets of HFO trainer show no influence to others. Additionally, the executable generated can run provided a common RoboCup simulation 2D environment.

More efficient action ingredients. HFO system provides some basic actions so users can pay more attention on how to make decisions, which gives an environment to implement of Reinforcement Learning. We also provide some atomic actions alike, while optimized and verified to be more efficient. **Fig. 3** shows cycle spent on catching ball between them.

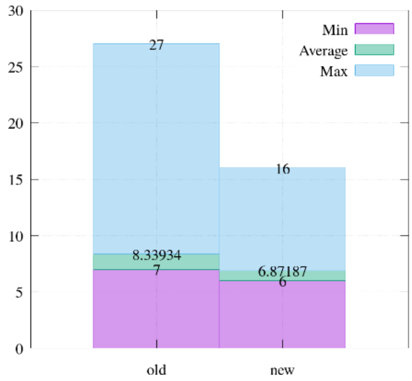

Fig. 3. Improvement of catching ball in average

HFO with more conditions. The original HFO system defined a lot of constants like HFO area and others in the modified rcssserver. After transforming them into a trainer, it's much easier to modified them and we extended variables to make it more flexible.

3.5 Application and future work

Automatic comparison between different teams. Fig. 4 shows how we evaluate different methods of catching a ball using HFO Trainer.

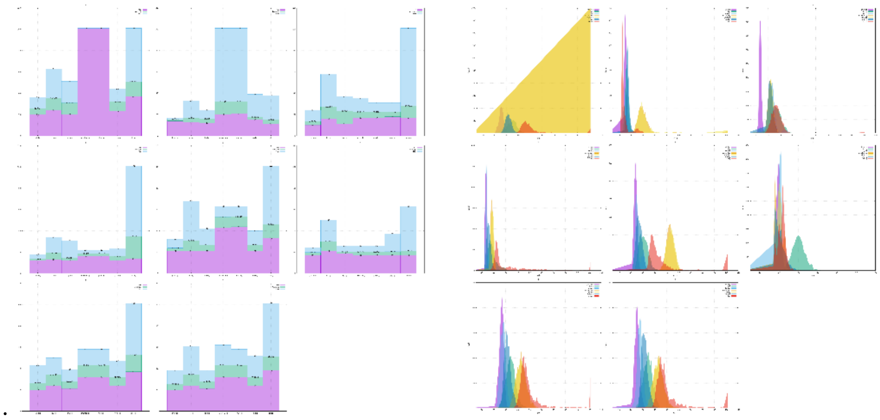

Fig. 4. Key data and frequency distribution of different methods

Preparation to combination with Machine Learning platform. HFO is designed to do research on Reinforcement Learning[12][13], so the trainer needs an API to run with platforms like TensorFlow or Python programs.

4 Expectation

We've noticed there have been many excellent teams making great effort to attempt to solving multi-agents problem and some of them has made exciting work. We are delighted and hopeful to explore the better method on it with previous experience, thoughts from everyone and intelligence of ourselves.

In the future, our focus will be divided into several parts below.

Strengthen the offensive performance. Our attacking players are often in trouble in attack and it is difficult to form effective cooperation. Next, we will continue to optimize the code and improve the performance of attackers.

Machine learning related content. We have partially introduced reinforcement learning and supervised learning to assist our work, but due to lack of experience, we still need some time to further optimize our players.

Code review. We hope that by refactoring the code, we can finally make these codes easy to read and optimize again.

References

[1] RoboCup tools agent2d. https://zh.osdn.net/projects/rctools/

[2] Akiyama, H., Nakashima, T.: Helios base: An open source package for the RoboCup soccer 2d simulation (01 2014). https://doi.org/10.1007/978-3-662-44468-946

[3] Milind Tambe, Jafar Adibi, Yaser Al-Onaizan, Ali Erdem, Gal A. Kaminka, Stacy C.Marsella, and Ion Muslea. Building agent teams using an explicit teamwork model and learning[J]. Artificial Intelligence, 111(1):215–239, 1999.

[4] Nuno Lau, Luis Seabra Lopes, Gustavo Corrente, Nelson Filipe, and Ricardo Sequeira. Robot team coordination using dynamic role and positioning assignment and role-based set plays[J]. Mechatronics, 21(2):445–454, 2011.

[5] Liping Cheng and Chuanxi Liu, Bo Yan "Improved Hierarchical A-star Algorithm for Optimal Parking Path Planning of the Large Parking Lot"; Proceeding of the IEEE International Conference on Information and Automation Hailar, China, July 2014.

[6] Elisabete Fernandes, Pedro Costa, Jośe Lima, Germano Veiga "Towards an Orientation Enhanced A star Algorithm for Robotic Navigation"; 2015 IEEE.

[7] Zhou Weiteng, Han Baoming, Li Dewei, Zheng Bin "Improved Reversely A star Path Search Algorithm based on the Comparison in Valuation of Shared Neighbor Nodes"; 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP) June 9-11, 2013, Beijing, China.

[8] Jan Murray, Frieder Stolzenburg, and Toshiaki Arai. Hybrid state machines with timed synchronization for multi-robot system specification[J]. KI, 3/06:45–50, 2006.

[9] Mota, L.; Lau, N.; Reis, L.P.; Co-ordination in RoboCup's 2D simulation league: Set plays as flexible, multi- robot plans[C]. 2010 IEEE Conf. on Robotics, Automation and Mechatronics, RAM2010, pp. 362-367.

[10] Mota, L.; Reis, L.P.; A Common Framework for Cooperative Robotics: An Open, Fault Tolerant Architecture for Multi-league RoboCup Teams[C]. Int. *Conf. Simulation Modeling and Progr. for Aut. Robots* (SIMPAR).

[11] Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half Field Offense in RoboCup Soccer: A Multiagent Reinforcement Learning Case Study; *RoboCup International Symposium 2006*.

[12] Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half Field Offense in RoboCup Soccer: A Multiagent Reinforcement Learning Case Study. In *RoboCup-2006: Robot Soccer World Cup X*, 2006.

[13] Matthew Hausknecht, Peter Stone. Deep Reinforcement Learning in Parameterized Action Space. ICLR 2016.