HELIOS2021: Team Description Paper

Masaki Yamaguchi[1], Ryota Kuga[1], Hiroki Omori[1],
Takuya Fukushima[1], Tomoharu Nakashima[1], and Hidehisa Akiyama[2]

[1]Osaka Prefecture University, Osaka, Japan
{masaki.yamaguchi,ryota.kuga,hiroki.omori}@kis.osakafu-u.ac.jp
{takuya.fukushima,tomoharu.nakashima}@kis.osakafu-u.ac.jp
[2]Okayama University of Science, Okayama, Japan
akiyama@mis.ous.ac.jp

Abstract. This team description paper introduces the overview of the previous works and the recent research themes of Team HELIOS2021. We have been working on the improvement of statistical team performance indicators (e.g., winning rate and the average of scored goals) by considering the difference in the abilities between our and opponent's player agents. We propose a method that takes such ability-difference into account to swap the positions within our team. We also present an team evaluation system which has a client server architecture, which automatically perform a lot of games and analyze the game results through the interaction with a human operator. This team evaluation system can also be extended to develop an online competition manager.

1 Introduction

Team HELIOS2021 has participated in the RoboCup competition since 2000, and has won four championships [1]. The team has never failed to be one of the top four teams since the year 2005.

This team description paper presents our current efforts in developing the team. We recently focus on the position exchange by considering the matchup between players. The remainder of this paper is organized as follows. Section 2 describes the overview of our previous works. Section 3 describes our recent approach of position exchange by considering heterogeneous player agents. Section 4 describes our recent approach of team performance evaluation. Section 5 provides some conclusions.

2 Previous Works

We have released a part of our team's source codes and their related debugging tools in order to help new teams to participate in the competitions and to start the research of multiagent systems [2]. Currently, the released software packages are available at our project site[1].

[1] https://osdn.net/projects/rctools/ (Please cite [2] when you publish papers using the software packages in this site.)

We have proposed two important methods for developing a (simulated) robotic soccer team, a formation model using triangulation [3] and a framework of action sequence planning [4]. These methods have already been implemented in the released software so that it allows us to develop a working simulated soccer team effortlessly. Acquiring an effective evaluation function for action sequence planning is still a problem to be solved. We are trying to apply some machine learning methods for this problem [5,6]. Team Receptivity also focused on improving the agent's decision evaluation capabilities by using a neural network [7].

In the soccer simulation 2D league, one of the essential tasks in the development of a team is to switch our strategy to more effective one according to the opponent's strategy. The sooner the opponent team's strategy is identified, the more chances to win the game the team can obtain by adapting its strategy. Thus, we have also been working on the analysis of team strategies from different points of perspectives. For example, an approach of strategy analysis is an opponent formation identification [8]. Moreover, we have proposed team-analysis methods using action commands [9,10]. Team Fractals2019 [11] and Team Titans [12] tackled to an assignment problem of heterogeneous players in order to improve team performance by using search algorithms. These approaches focus on the team strategies, not between individual players. On the other hand, since the last year we have been working on the matchup between individual players. The next section provides such approach and gives an idea to utilize the matchup between individual players.

3 Position Exchange based on Player's MatchUp

It is common that a player confronts its opponent players during the game. The pair of such players between a player from one team and its confronting player from the other team is called "matchup" in this paper. A matchup is preferable when our player's ability (e.g., dash power) is higher than that of its confronting player. This year we consider to make as many preferable matchup as possible during a game by exchanging the position of our players. There are two steps in the position exchange: One is to identify the matchup of each player as soon as possible during a game, and the other is to determine which players should exchange to make more preferable matchups.

3.1 Proposed Method

For the position exchange, we propose a method for making as many preferable matchups as possible during an ongoing game. The proposed method consists of two steps. One is "matchup," and the other is "position exchange". The following subsections gives each step of the proposed method.

Identifying matchups We define "matchup" as a pair of our player and its confronting opponent player. Since we do not know the opponent strategy neither their formation, it is not possible to identify the matchups for a game beforehand. The results of such identification should be used in the following position-exchange step. Thus, it is necessary to identify the matchups as soon as possible once a game starts. We consider that our player and its nearest opponent player forms a matchup. Note that it is not necessary for a pair of our and opponent's players in the identified matchups to be a one-to-one correspondence.

Position exchange Position exchange is executed by considering the difference in player's abilities within the identified matchups. The positions of two players in our team are swapped. The two players are selected by choosing two matchups that are not preferable but would become more preferable if our players are exchanged. Note that reserve players are not considered in the position exchange. The criterion for choosing two matchups is heuristically determined by a human developer.

3.2 Experiments

In order to evaluate the performance of the proposed position exchange method, we conducted computational experiments using Team HELIOS2019. The soccer player presents a set of heterogeneous parameters that characterizes the players' ability. Each set of parameters is called *player type*. The coach agent is responsible to determine which player has which player type before starting a soccer game. Among the presented player types, we assign the default player type to the goalie. For the remaining ten players, we first extract ten player types that have a high running ability. The running ability is measured as the sprint time of 30 *meter* dash. The selected ten player types are then randomly assigned to the remaining players (i.e., ten players excluding the goalie). In addition, we divide a formation into three roles: Forward, midfielder and defender. Based on the comparison of the differences in the running ability of $30m$ sprint within the matchup for each of the three roles, our player's positions are exchanged based on the following rules.

- *all*: Positions are exchanged if the order of our players in one of the three roles with respect to the running ability is the same as the order of opponent players that is included in the matchups.
- *Specialized* (noted as *sp*): Positions are exchanged if the best running-ability player in our team within the roles is paired with the worst player in the opponent team. In *sp* setting, the other players are also exchanged in accordance with *all* in role group

Figure 1 shows an example of the proposed position exchange. In this figure, yellow players belong to the same role in our team and red players are their confronting opponent players in the identified matchups. The figure assumes a role group that consists of three players. The opponent players are shown in ascending order by the running ability from left to right. In our team, the best running ability (i.e., the sprint time for $30m$) is 33 cycles, the second is 35 cycles and the third is 40 cycles. In the opponent team, they are 34 cycles, 36 cycles

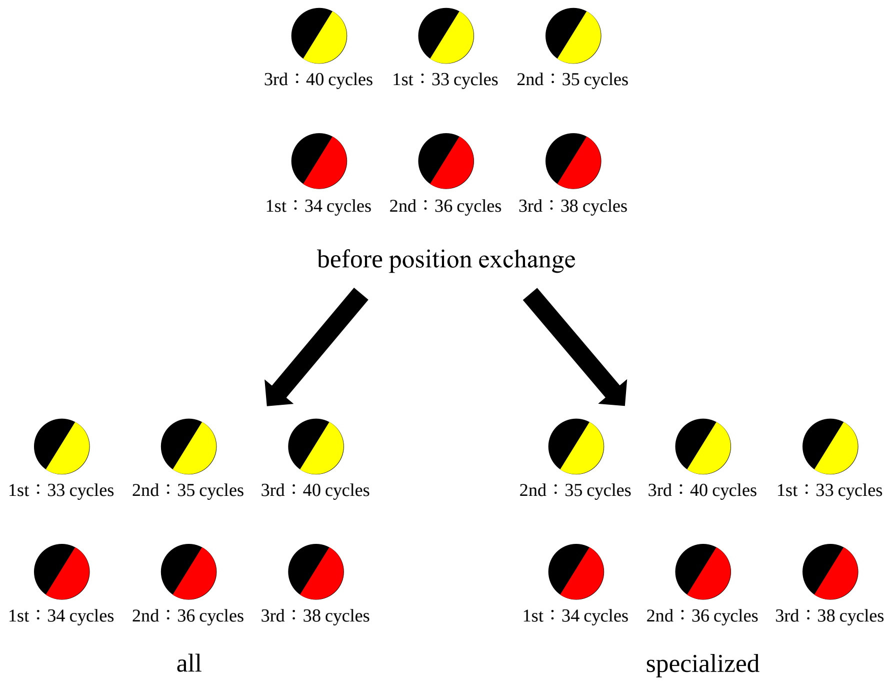

Fig. 1. Example of position exchange

and 38 cycles respectively. Each column means a matchup of our player and its confronting opponent player. This figure shows the comparison between the two position exchanges. As shown in the figure, the *all* rule assigns our players corresponding to the order of the running ability. On the other hand, the *sp* rule assigns our best player to the opponent worst player.

We conducted the experiments with following six settings:

- NOSWAP: no position exchange
- DF_all: exchange defender by *all*
- MF_all: exchange midfielder by *all*
- FW_all: exchange forward by *all*
- FW_sp: exchange forward by *sp*
- COMB: exchange by DF_all and FW_sp simultaneously

Position Exchange is executed once during a game at our setplay period after 1000 cycles. 1000 games are run for each setting. Random seed for heterogeneous player types is set as $seed = 10 \times i \, (i = 1, \ldots, N)$, where N is the number of the games.

Table 1 describes winning rate (%), the average of scored goals (Our score), the average of conceded goals (Opp score), the average number of our through-passes (# of our t-passes) and the average number of opponent's through-passes

(# of opp t-passes). This table shows that some Opp score and # of opp t-passes of DF_all and COMB are decreased from NOSWAP by a statistical test with the 5% level. We expect the distance between a matchup of our defender and opponent forward player is one of the important feature variables that affects these performance indicators. We are now investigating it in more detail.

Table 1. Game results of each experimental setting

Opp team	Setting	Win(%)	Our score	Opp score	# of our t-passes	# of opp t-passes
CYRUS	NOSWAP	66.3	1.471	0.371	3.551	2.967
	DF_all	60.2	1.426	0.481	3.520	3.130
	MF_all	63.2	1.404	0.375	3.624	2.867
	FW_all	60.4	1.419	0.441	3.480	2.874
	FW_sp	60.4	1.428	0.424	3.630	3.021
	COMB	64.7	1.431	0.407	3.723	3.120
Yushan	NOSWAP	70.5	1.834	0.564	5.184	5.289
	DF_all	72.0	1.813	0.482	4.972	4.595
	MF_all	70.2	1.817	0.562	5.140	5.145
	FW_all	72.5	1.860	0.514	5.142	5.124
	FW_sp	71.3	1.909	0.555	5.189	5.165
	COMB	69.2	1.728	0.467	4.960	4.492
HillStone	NOSWAP	96.9	4.122	0.217	9.527	2.628
	DF_all	96.5	4.095	0.195	9.437	2.747
	MF_all	97.2	3.992	0.216	9.480	2.601
	FW_all	96.1	3.680	0.206	9.009	2.673
	FW_sp	97.7	4.154	0.231	9.604	2.607
	COMB	96.7	4.055	0.199	9.425	2.653
Jyo_sen	NOSWAP	100.0	10.247	0.252	15.605	1.925
	DF_all	100.0	10.149	0.240	15.448	2.081
	MF_all	100.0	10.262	0.249	15.578	1.842
	FW_all	100.0	10.207	0.246	15.430	1.930
	FW_sp	100.0	10.317	0.236	15.664	1.941
	COMB	100.0	10.233	0.235	15.402	1.937
Shibaura	NOSWAP	100.0	9.680	0.090	15.389	1.727
	DF_all	99.9	9.609	0.117	15.154	1.680
	MF_all	99.9	9.677	0.104	15.427	1.674
	FW_all	100.0	9.535	0.105	15.249	1.651
	FW_sp	100.0	9.979	0.106	15.606	1.683
	COMB	100.0	9.692	0.088	15.346	1.702

4 Team Performance Evaluation

One of the problems in developing a team for soccer simulation 2D is evaluation of team performance. The game results are different game by game due to the randomness in the simulator and also in the teams even though the games were conducted with the same match-up. Thus, in order to evaluate the team performance as correctly as possible, a large amount of games are required to obtain statistical performance indicators. Due to the limitation of computational resources, the important requirement for an evaluation system is the effectiveness. Because even a single game requires a long sequence of commands to conduct, we need an easy-to-use system that automatically execute a large number of games, which allows the analysis of the game results.

Therefore, we developed an interactive performance evaluation system that consists of one server computer and other host computers. The server of the system assigns a host computer for executing requested games and aggregating their results. We also introduced a chat-bot interface that enables users to send a request to the server interactively. The Slack bot API [13] is used for this purpose. This interactive interface makes it easier to run evaluations from various devices through a Slack application. In addition, the aggregated results are recorded on the Google Sheets [14], which enables a share of the results always among all developers.

Figure 2 shows the overview of our performance evaluation system. The procedure of our evaluation system is as follows:

Step 1: Select game settings
Step 2: Assign computers
Step 3: Execute games respectively
Step 4: Analyze the Log files
Step 5: Write the game results

In Step. 1, we send game settings (the branch name in Git repository, the name of opponent team, the number of games, and so on) to the server computer through the Slack bot as a request. In Step. 2, in response to the message in Step 1, the server assigns games to host computers according to the availability of computational resource. The server searches for available host computers by checking the CPU usage. If a host computer returns a busy status, the server does not assign any task to that host computer to avoid a resource conflict. In Step. 3, each assigned host runs the soccer simulator according to the specified soccer games. The host computers also hold the game log files after the soccer games are finished. In Step. 4, The host computer analyzes the log files after finishing the game. The analyzed game results (goals scored, goals conceded) are saved in CSV format on the assigned host computers. Then the CSV files are transmitted to the server. In the final step, the game results are summarized to the Google Sheets.

This automatic evaluation system can be applied to a competition manager, which we are currently working on for the RoboCup 2021. While the team evaluation system provides a repeated process of conducting the game, the competition manager requires a schedule to conduct various games. Thus, the team evaluation system is modified so that games with any match-up with a set of teams are conducted according to a pre-specified schedule. As the competition manager needs a binary test function where each team can check if it plays a game properly without any problems, the modified team evaluation system accepts the team binary upload from a designated user accounts through the tournament Slack workspace. Log files are copied to a shared directory (e.g., Dropbox and

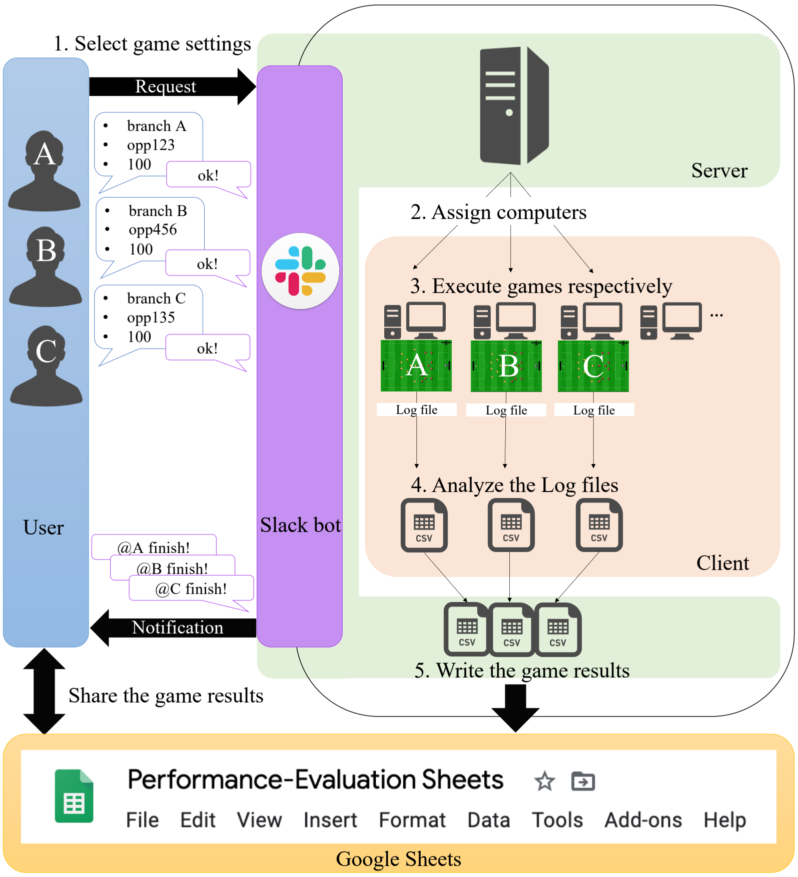

Fig. 2. Overview of our performance evaluation system

Amazon S3) so that teams can re-play the test games. The competition manager will be developed by the time of RoboCup 2021 tournaments in July 2021.

5 Conclusion

This paper described the previous efforts and the current research topics of team HELIOS2021. We are currently trying to improve the team performance by considering the difference among the players' abilities. We proposed a position exchange approach in order to optimize the role assignment of heterogeneous players during a game. A series of computational experiments are conducted in order to show the effectiveness of the position change. In addition, we presented a system to evaluate the team performance that automates all the necessary process for iterating soccer games and summarize their results. Our evaluation system helps us to improve a team based on the statistical results, through the interactive application interface.

References

1. Hidehisa Akiyama, Tomoharu Nakashima, "HELIOS2018: RoboCup 2018 Soccer Simulation 2D League Champion", *RoboCup 2018: Robot Soccer World Cup XVII*, pp. 18–22, 2018.
2. Hidehisa Akiyama, Tomoharu Nakashima, "HELIOS Base: An Open Source Package for the Robocup Soccer 2D Simulation", *RoboCup 2013: Robot World Cup XVII.*, pp. 528–535, 2014.
3. Hidehisa Akiyama, Itsuki Noda, "Multi-Agent Positioning Mechanism in the Dynamic Environment", *RoboCup 2007: Robot Soccer World Cup XI*, pp. 377–384, 2008.
4. Hidehisa Akiyama, Shigeto Aramaki, Tomoharu Nakashima, "Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation", *Proc. of 4th IEEE international Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 170-177, 2012.
5. Hidehisa Akiyama, Masashi Fukuyado, Toshihiro Gochou, Shigeto Aramaki, "Learning Evaluation Function for RoboCup Soccer Simulation using Humans' Choice", *Proceedings of SCIS & ISIS 2018*, 2018.
6. Takuya Fukushima, Tomoharu Nakashima, Hidehisa Akiyama, "Mimicking an Expert Team through the Learning of Evaluation Functions from Action Sequences", *RoboCup 2018: Robot World Cup XXII*, pp.170-180, 2019.
7. Mike Li, "Receptivity: Team Description Paper 2018 Fine Tuning of Agent Decision Evaluation", *RoboCup 2019 Team Description Paper*, 6 pages, 2019.
8. Takuya Fukushima, Tomoharu Nakashima, Hidehisa Akiyama, "Online Opponent Formation Identification Based on Position Information", *RoboCup 2017: Robot World Cup XXI.*, pp. 241–251, 2018.
9. Jiarun Zhong, Tomoharu Nakashima, Hidehisa Akiyama, "A Study on the Analysis of Soccer Games Using Distributed Representation of Actions and Players", *ICIC Express Letters*, Vol. 13, No. 4, pp. 303–310, 2019.
10. Takuya Fukushima, Tomoharu Nakashima, Hidehisa Akiyama, "Similarity Analysis of Action Trajectories Based on Kick Distributions", *RoboCup 2019: Robot World Cup XXIII*, pp. 58–70, 2019.
11. Mikhail Prokopenko and Peter Wang, "Fractals2019: Combinatorial Optimisation with Dynamic Constraint Annealing", *RoboCup 2019: Robot World Cup XXIII*, pp. 616–630, 2019.
12. Alexandre F. Costa, Giovany F. Teixeira, Maykel Rodrigues, "Titans of Robotics Soccer Simulation 2D", *RoboCup 2019 Team Description Paper*, 6 pages, 2019.
13. Slack API, `https://api.slack.com`. Last accessed 1 Mar 2020.
14. Google Sheets, `https://www.google.com/sheets/about`. Last accessed 1 Mar 2020.