

# HELIOS2018: Team Description Paper

Tomoharu Nakashima<sup>1</sup>, Hidehisa Akiyama<sup>2</sup>, Yudai Suzuki<sup>1</sup>, An Ohori<sup>1</sup>, and Takuya Fukushima<sup>1</sup>

<sup>1</sup>Osaka Prefecture University, Osaka, Japan  
tomoharu.nakashima@kis.osakafu-u.ac.jp  
takuya.fukushima@edu.osakafu-u.ac.jp  
sya01138@edu.osakafu-u.ac.jp  
sya01053@edu.osakafu-u.ac.jp

<sup>2</sup>Fukuoka University, Fukuoka, Japan  
akym@fukuoka-u.ac.jp

**Abstract.** This team description paper explains a task that Team HELIOS2018 is tackling this year as well as its general introduction. The task of the team this year is to share the knowledge between players in terms of action sequence planning in order to increase the achievement probability of planned action sequences. Knowledge sharing in this TDP is defined as having a prediction model of a ball kicker by a non-kicker. The increase in the achievement probability of action sequences is expected to help execute the action sequence as planned by the ball kicker. The usefulness of the idea is investigated through a series of computational experiments.

**Keywords:** RoboCup · Knowledge sharing · Action sequence planning

## 1 Introduction

HELIOS2018 is a simulated soccer team for the RoboCup soccer 2D simulation league. The team has been participating in the RoboCup competition since 2000, and has won three championships [1]. The team has released a part of their source codes and related debugging tools in order to help new teams to participate in the competitions and to start the research of multiagent systems.

In recent years, we have focused on the improvement of action sequence planning using a tree search mentioned. This paper explains our recent approach in order to increase the achievement probability of planned action sequences.

## 2 Previous Works

### 2.1 Formation Model using Triangulation

We have proposed a positioning mechanism that uses Delaunay Triangulation and the linear interpolation algorithm [3]. This model divides the 2D soccer field into several triangles based on given training data. Each training data

affects only the divided region where it belongs to. This feature enables us to adjust the players' position locally. To compose training data set, we use human's observation and their intuitive input. The formation editor, `fedit2`, enables us to do such operations.

## 2.2 Framework for Action Sequence Planning

We have proposed a tree search framework to generate and evaluate action sequences performed by multiple simulated soccer players [4]. This framework searches for the best sequence of actions using a tree-structured candidate action generator and an evaluation function of the candidate actions.

## 2.3 Released Software

We have released several open source software that help us to develop a simulated soccer team [2]. The proposed team formation model and framework for action sequence planning have already been available in the released code. Several ideas were inspired from other released codes[5–10]. We would like to thank their effort and contributions. The following packages are available so far<sup>1</sup>:

- `librcsc`: a base library for developing a simulated soccer team.
- `agent2d(HELIOS Base)`: a sample team code.
- `soccerwindow2`: a high functional viewer, which can be used as a monitor client, a log player and a visual debugger.
- `fedit2`: a formation editor for `agent2d`.

# 3 Knowledge Sharing of Action Sequence Planning

## 3.1 Action Sequence Planning

As described in 2.2, the team has employed an action sequence planning framework. This planning is made by a ball kicker (called a kicker hereafter). First, the candidate actions are generated from the current situation of the soccer field. Each action is assigned an evaluation value that represents the quality of the action. The selected action is then used as a second kicking point to generate further candidate actions. This process expands the sequence of candidate actions in a tree form. Best-first search algorithm is used to traverse a tree and to expand nodes.

An example of the planning process is depicted in Fig. 1. In Fig. 1, the kicker generates three candidates for the first action (i.e., pass, pass, and dribble). Each of the three action has an evaluation value in the corresponding node. That is, the evaluation value of the first pass is 30, the second pass is 20, and the dribble is 15. In this case, the first pass with the highest evaluation value is employed as the first action. Further candidate actions are generated from the selected

<sup>1</sup> Available at: <http://osdn.net/projects/rctools/>

pass action. We call the level of the tree as depth of the action sequence. Two actions (pass and dribble) in Depth 2 are added as the candidate action with the corresponding evaluation values. The action sequence is updated as the one with the highest evaluation values among the candidate. In this case, the pass in Depth 2 is selected as it has the highest evaluation of 35. Thus the resultant action sequence is “pass-pass”.

The problem here is that even though the generated chain action by the above process is good, its execution is not easy. Most of the times, the generated action sequence is intervened during the course of its execution and it is necessary to plan the next action sequence again. One reason for this is that non-ball-kicking players (we call this non-kickers hereafter) do not know the plan of the kicker, which leads the incomplete formation to perfectly execute the plan. In this TDP, we tackle this problem using the idea of knowledge sharing.

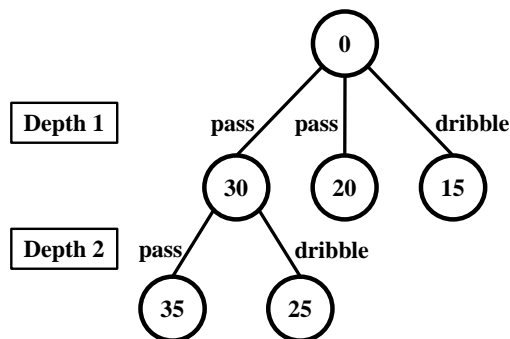


Fig. 1. Action sequence planning.

### 3.2 Knowledge Sharing

In order to tackle the problem described in Subsection 3.1, knowledge sharing is employed. Knowledge sharing is to have a common knowledge among different players. In the context of the action sequence planning in this TDP, the knowledge sharing means that the action sequence generated by non-kickers are made as close to the one planned by the kicker as possible. To do so, non-kickers focus on obtaining the information around the kicker as much as possible. Then they try to generate the same action sequence as that by the kicker. The non-kicker selects the next action considering the action sequence which will be executed from the current situation.

## 4 Computational Experiments

In order to examine the effectiveness of knowledge sharing, we have done a preliminary experiments using Team opuSCOM, which is based on agent2d. Knowl-

edge sharing is implemented in the opuSCOM players so that all the players are able to expect the action sequence of a kicker when they are non-kickers. In particular, we focus on a situation depicted in Fig. 2. As shown in the figure, action sequences with the depth of less than or equal to two are only considered. Furthermore, it is assumed that all the actions in the plan are passes. This is because it is easier to measure the effectiveness of the knowledge sharing without considering the presence of dribbles, which are difficult to differentiate with self passes.

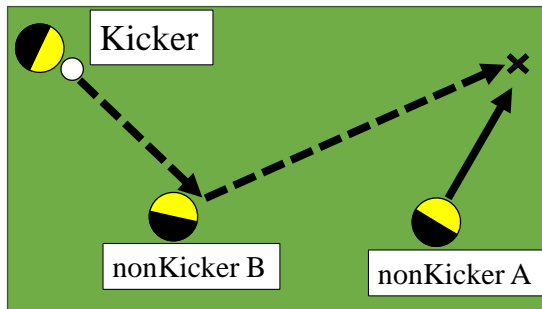


Fig. 2. Experimental situation.

Table 1. Success rates of the knowledge sharing.

Opponents	#Planning	Sharing Rate(%)
agent2d	95	89.40
HillStone	89	89.64
Esperanza	84	90.16
Toyosu-Galaxy	30	92.20
WIT	51	89.49
Fifty-Storms	100	90.48
ITAndroids	84	89.81
PersianGulf2017	73	87.45
Ri-one2017	65	89.72
opuSCOM (Before)	114	90.29

Using ten teams that participated in the previous domestic as well as international competitions, the execution rates of the planned action sequences are measured when knowledge sharing is implemented in Team opuSCOM. First, we show the success rates of the knowledge sharing in Table 1. The table shows the average number of planned chain actions as well as successfully shared planned

**Table 2.** Successful execution rates of planned chain actions.

Opponents		#Planning	D1 (%)	D2 (%)	D2 per D1 (%)
agent2d	Before	101	73.37	42.12	57.40
	After	95	73.39	47.15	64.25
HillStone	Before	94	71.92	39.21	54.52
	After	89	70.20	45.79	65.23
Esperanza	Before	89	72.49	43.53	60.05
	After	84	73.34	48.92	66.70
Toyosu-Galaxy	Before	29	74.46	30.50	40.96
	After	30	75.41	37.12	49.22
WIT	Before	51	74.06	44.36	59.90
	After	51	73.57	47.56	64.65
FiftyStorms	Before	104	67.80	37.52	55.34
	After	100	68.73	43.01	62.58
ITAndroids	Before	91	73.63	35.61	48.36
	After	84	73.95	40.65	54.97
PersianGulf2017	Before	71	70.29	39.74	56.54
	After	73	68.67	40.25	58.61
Ri-one2017	Before	67	73.71	36.00	48.84
	After	65	73.06	37.03	50.68
opuSCOM (Before)	Before	120	71.13	41.04	57.70
	After	114	72.43	47.14	65.10

chain actions in 300 games for each opponent team. From this table, we can see that knowledge sharing is achieved with high precision.

Next, we investigate the execution of the planned action sequences. Table 2 shows the average number of planned action sequences (#Planning), the successful execution rates of the planned action sequences for the depth of one (D1) and two (D2), and the rate of successful execution of the depth-two action sequences over the successful depth-one action sequences (D2 per D1). Table 2 also compares the performance of the two versions of opuSCOM, one without the knowledge sharing (before) and the other with it (after). From this table, we can see that the successful execution was increased by the introduction of knowledge sharing. The difference in the performance between the team with the knowledge sharing and the conventional one is significant by a statistical test with the 5% level.

## 5 Conclusion

This team description paper described the approach of the team towards the successful execution of the planned action sequences. Knowledge sharing is introduced for this purpose. A series of computational experiments were conducted in order to show the effectiveness of the approach.

## References

1. Hidehisa Akiyama, Tomoharu Nakashima, “HELIOS2012: RoboCup 2012 Soccer Simulation 2D League Champion”, *RoboCup 2012: Robot Soccer World Cup XVI*, pp. 13–19, 2013
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 Co-operative 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. Peter Stone, Patrick Riley, Manuela Veloso, “The CMUnited-99 Champion Simulator Team”, *RoboCup-99: Robot Soccer World Cup III*, pp. 35–48, 2000.
6. Luis Paulo Reis, Nuno Lau, Eugenio Olivéira, “Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team”, *Balancing Reactivity and Social Deliberation in MAS*, pp. 175–197, 2001.
7. Koji Nakayama, Takuya Ako, Toshihiro Suzuki, Ikuo Takeuchi, “Team YowAI-2002”, *RoboCup 2002: Robot Soccer World Cup VI*, 2002.
8. Jinyi Yao, Jiang Chen, Yunpeng Cai, Shi Li, “Architecture of TsinghuAeolus”, *RoboCup 2001: Robot Soccer World Cup V*, pp. 491–494, 2002.
9. Jelle R. Kok, Nikos Vlassis, F.C.A. Groen, “UvA Trilearn 2003 team description”, *Proceedings CD RoboCup 2003*, 2003.
10. M. Riedmiller, T. Gabel, J. Knabe, and H. Strasdat, “Brainstormers 2D - Team Description 2005”, *Proceedings CD RoboCup 2005*, 2005.