

The Characteristics of the Card Game Daihinmin

Mitsuo Wakatsuki^{*}, Seiya Okubo[†], Yuta Kado^{*},
Yamato Takeuchi^{*}, Tetsuro Nishino^{*}

Abstract

Combinatorial game theory has many ways of measuring game complexity. This study presents several measures of complexity for the card game Daihinmin to obtain some of its characteristics. Through computer experiments using programs playing Daihinmin, we collect data on the average number of possible moves and the average length of game in the case of the standard rules and in that of the rules including some local rules such as 11-Back. Then, we calculate some indicators such as the strategic complexity of the game and investigate the characteristics of these indicators. Furthermore, a questionnaire survey is conducted to find out which move a human will play from the initial hand in the specified situation. From moves made by various computer programs in the same situation, we show the difference in playing styles between humans and computers.

Keywords: Daihinmin, game complexity, game play style, game with imperfect information

1 Introduction

People tend to use different criteria when defining a good card game. Some people find that a game that has simple mechanics can quickly become boring, while others argue that the complexity of a game's mechanics creates great strategic potential.

Combinatorial game theory measures game complexity in several ways. For example, a game tree size is the total number of possible games that can be played. In this case, it is the number of leaf nodes in the game tree rooted at the game's initial position. The game tree is typically significantly larger than the state space because the same positions can occur in many games by making moves in a different order. The indicators including the game tree size for representative games with perfect information, such as Chess and Shogi, are relatively easy to analyze and are well known [9]. In contrast, there is a lack of research on how to analyze the indicators for multiplayer imperfect information games, such as card games. In this study, we focus on a card game called Daihinmin. Programs that play Daihinmin on a computer include those that use heuristics based on human thinking to determine the next hand [12, 13] and those that use machine learning such as the Monte Carlo method [6, 11]. Other studies of computer Daihinmin include the

^{*} The University of Electro-Communications, Tokyo, Japan

[†] University of Shizuoka, Shizuoka, Japan

analysis of the behavior of programs using decision trees [3, 4], the extraction of the characteristics of programs using n-gram and classify them [7, 8], the characteristics of Daihinmin from program logs [9], and the investigation of the effect of evaluation values in the Monte Carlo method [10, 15]. This study presents several measures of the complexity of the card game Daihinmin for the first time to obtain several of its characteristics.

On the other hand, for people to play card games on a computer without getting bored, it is necessary to develop a game program close to human thinking. Are the playing styles of card game engines similar to those of humans at present? It would appear that humans impose unwritten rules on card games. The Daihinmin program kou [12, 13] uses a heuristic approach based on human thinking. Using Marvin Minsky's Society of Mind theory [5], we constructed a model of the minds of game players [14]. In this study, for that purpose, we present the differences in playing styles of Daihinmin between humans and computers.

2 Preliminaries

2.1 Computer Daihinmin

Daihinmin (Grand Pauper), which is also called Daifugo (Grand Millionaire) [1], is a card game mainly played in Japan, although similar games are played around the world. It is a multiplayer game of imperfect information that has been extensively studied in recent years. Computer Daihinmin refers to playing the game on a computer. The UEC Computer Daihinmin Convention (UECda) [2] is an annual competition for Daihinmin computer programs. Several thousand games are played in the competition annually by computers with extreme computational capabilities. The competing algorithms seek to win over the course of many games, without the influence of an initial hand.

This study adopts the framework utilized by the UECda. While there are numerous sets of localized Daihinmin rules, the UECda implements the specific rules outlined in Okubo et al. [9]. We also use the following localized Daihinmin rules.

11-Back (Jack-Back): When a player plays a hand containing a jack, a temporary revolution occurs that only lasts for the current round.

6-Reverse: When a player plays a hand containing a 6, the order of the turn is reversed until the end of the game.

5-Skip (5 Tobi): When a player plays a hand containing a 5, the next player is skipped.

2.2 Daihinmin Programs

There have been many studies involving Daihinmin that require (and result in) the creation of strong programs [3]-[4], [6]-[15]; we used the following programs.

Default: This program is the standard Daihinmin program on the official website of UECda [2]. It is a simple heuristic program, and it plays the weakest card(s) that can be played.

Ishinomaki: This program participated in UECda-2011 and it partially utilizes the Monte Carlo method.

Snowl: This program was developed by Fumiya Suto and is characterized by machine learning and the Monte Carlo method [11]. Snowl won the UECda-2010 and has been used in many studies.

Crow: This program was developed by Satoshi Konuma. It utilizes an ε -greedy method to find the best possible move while applying the Monte Carlo method. Crow won the UECda-2011.

Paoon: This program also utilizes the Monte Carlo method. Paoon won the open-weight class of the UECda-2012.

Wisteria: This program was developed by Katsuki Ohto and uses supervised learning of policy functions based on policy gradients, as well as the Monte Carlo method [6]. Wisteria won the open-weight class of the UECda-2015.

3 Effects of Localized Daihinmin Rules

3.1 Universal Estimates

Universal estimates are one type of indicator used to characterize games. For games in which look-ahead search algorithms are important, the size of the game tree is an important indicator of its difficulty. The size of the game tree is defined as B^D for an average number of possible moves B and average game length D . The strategic complexity of a game can be expressed as \sqrt{B}/D . Universal estimates for representative games, such as Chess and Shogi, are shown in Table 1 of Okubo et al. [9].

To obtain universal estimates for Daihinmin, it is necessary to calculate the size of the game tree. Although a complete game tree cannot be created for Daihinmin because it is a multiplayer imperfect information game, we can investigate B and D through computer experiments. Here, the issue is how to handle passes when evaluating average game length D . We consider that D contains the number of passes. In this manner, we can calculate the upper bound of B from a player's initial hand. For further details see Okubo et al. [9].

3.2 Evaluation of Effects of 11-Back

Through computer experiments, we collected data on the average number of possible moves B , average game length D , and history of ranking in the cases of the UECda standard rules and the 11-Back rules, which are the UECda standard rules including 11-Back. In addition, to investigate the cause of the differences between these cases, we calculated the number of passes per game $\#pass$, average game length when the top rank D_{top} , and average number of possible moves when "the field is lead" B_{lead} . Here, "the field is lead" means that there are no cards on the table.

The experiment consisted of 100 sets in which each set of 3,000 games was played by the same 5 programs. We used the Default, Ishinomaki, and Snowl Daihinmin programs in this experiment. In addition, we rewrote Default, Ishinomaki, and Snowl so that these programs could be used for 11-Back. We called these programs B11def, B11monte, and B11snowl, respectively.

Table 1 shows D , B , \sqrt{B}/D , D_{top} , $\#pass$, and B_{lead} for these experiments. Table 2 presents the difference of transition probabilities for each pair of two Special Titles between the case of the 11-Back rules and that of the UECda standard rules.

In Table 1, comparing the average game length D , we see that the length for Default is almost the same as that for B11def, while the length for B11monte is longer than that for Ishinomaki.

This is because Ishinomaki has more passes than Default, and B11monte judged that the pass is valid for 11-Back. Comparing the average number of possible moves B , we see that the number for Default is almost the same as that for B11def, while the number for B11monte is less than that for Ishinomaki. Therefore, 11-Back has no effect on increasing the average number of possible moves.

Table 1: Universal Estimates for Daihinmin

Player	D	B	\sqrt{B}/D	D_{top}	#pass	B_{lead}
Default	78.93	2.88	0.022	52.40	43.98	7.22
Ishinomaki	84.02	3.15	0.022	44.21	47.98	7.28
B11def	78.62	2.90	0.022	53.48	43.55	7.13
B11monte	86.07	2.94	0.020	48.87	50.13	7.13

From Table 2, in the cases of B11def and Default, we see that they are more likely to become a lower title than Heimin in the next game for Daifugo and Fugo with the 11-Back rules, compared to the UECda standard rules. On the other hand, with the 11-Back rules, they are more likely to become a higher title than Fugo in the next game for Daihinmin and Hinmin. We also conducted similar experiments in the cases of B11monte and Ishinomaki and found that Heimin is more likely to become not only Daifugo but also Daihinmin in the next game (For details see Wakatsuki et al. [16]). Therefore, 11-Back has the effect of reducing the gap between the titles.

Table 2: Difference of Transition Probabilities for Each Pair (B11def-Default [%])

Titles	Daifugo	Fugo	Heimin	Hinmin	Daihinmin
Daifugo	-7.47	-2.18	1.54	3.61	4.52
Fugo	-2.38	-2.20	0.05	1.81	2.71
Heimin	1.66	-0.06	-1.12	-0.47	-0.02
Hinmin	3.16	1.80	-0.12	-2.14	-2.70
Daihinmin	5.03	2.63	-0.34	-2.80	-4.52

Row: Current Title, Column: Next Title

3.3 Evaluation of Effects of 6-Reverse

Through computer experiments, we collected data on scores for each seat order in both cases of the UECda standard rules and that of the 6-Reverse rules, which are the UECda standard rules including 6-Reverse. In this experiment, we did not change the seats. In addition, to investigate the cause of the differences between these cases, we collected the number of times a 6 is played in all the games and the number of times a 6 is played until two players played all their cards (hereafter, we call this period “the early stage of the game”). Furthermore, considering the fact that the difference in the initial hands due to the exchange rule affects the number of times a 6 is played, we collected data for both the cases with and without the exchange rule.

The experiment consisted of 100 sets in which each set of 3,000 games was played by one strong program and four weaker programs. We used the Default, Snowl, and Paoon Daihinmin programs in this experiment, where Default is much weaker than Snowl, and Snowl is weaker than Paoon. In addition, we rewrote Default, Snowl, and Paoon so that these programs could be used for 6-Reverse. We called these programs R6def, R6snowl, and R6paoon, respectively.

Tables 3 and 4 show the average scores played by one Snowl (or R6snowl) and four Defaults (or R6defs) for the UECda standard rules $Score_{UECda}$ and the 6-Reverse rules $Score_{6Rev.}$, with and without the exchange rule, respectively. These tables also show the average number of times a 6 is played in all the games $\#6R_{all}$ and the average number of times a 6 is played in the early stage of the game $\#6R_{early}$ in the case of the 6-Reverse rules. Here, the number in parenthesis shows the seat order.

From Tables 3 and 4, the 6-Reverse rules tend give a higher ranking for the player with the seat order 4 (that is, Default(4)) compared to the UECda standard rules. We also conducted similar experiments played by one Paoon (or R6paoon) and four Snowls (or R6snowls), both with and without the exchange rule, respectively (For details see Wakatsuki et al. [16]). Comparing these results, we found that the average scores of Default(4) in the case of the 6-Reverse rules are much higher than those in the case of the UECda standard rules. This is because the difference in program strength between Snowl and Default is much larger than that between Paoon and Snowl. Therefore, the exchange rule eliminates the weaker cards (such as 3 and 4) from the initial hand of a strong program, and as a result, a strong program played a 6 in the early stage.

Table 3: Average Scores Played by One Snowl and Four Defaults with the Exchange Rule

Program	$Score_{UECda}$	$Score_{6Rev.}$	$\#6R_{early}$	$\#6R_{all}$
Snowl(0)	12878.02	13001.90	1170.00	1341.81
Default(1)	8269.99	8074.02	1332.54	1830.04
Default(2)	7986.04	7922.03	1322.21	1834.26
Default(3)	7962.82	7932.36	1249.18	1750.71
Default(4)	7903.13	8069.70	1256.79	1745.46

Table 4: Average Scores Played by One Snowl and Four Defaults without the Exchange Rule

Program	$Score_{UECda}$	$Score_{6Rev.}$	$\#6R_{early}$	$\#6R_{all}$
Snowl(0)	11593.05	11438.93	1519.79	1855.54
Default(1)	8602.30	8470.43	1421.46	1770.08
Default(2)	8298.88	8314.62	1383.10	1763.52
Default(3)	8271.07	8323.75	1338.70	1717.06
Default(4)	8219.71	8452.27	1364.88	1718.28

3.4 Evaluation of Effects of 5-Skip

Through computer experiments, we collected data on scores for each seat order in both the case of the UECda standard rules and that of the 5-Skip rules, which are the UECda standard rules including 5-Skip. In this experiment, we did not change the seats. Similar to the 6-Reverse rules, we collected the number of times a 5 is played in all the games and the number of times a 5 is played in the early stage. Furthermore, we collected data for both the cases, both with and without the exchange rule.

The experiment consisted of 100 sets in which each set of 3,000 games was played by one strong program and four weaker programs. We used the Default, Snowl, and Paoon Daihinmin programs in this experiment. In addition, we rewrote Default, Snowl, and Paoon so that these

programs could be used for 5-Skip, and we called these programs S5def, S5snowl, and S5paoon, respectively.

Tables 5 and 6 show the average scores played by one Snowl (or S5snowl) and four Defaults (or S5defs) for the UECda standard rules $Score_{UECda}$ and the 5-Skip rules $Score_{5Skip}$, with and without the exchange rule, respectively. These tables also show the average numbers of times a 5 is played in all the games $\#5S_{all}$ and the average number of times a 5 is played in the early stage $\#5S_{early}$ in the case of the 5-Skip rules.

We also conducted similar experiments played by one Paoon (or S5paoon) and four Snowls (or S5snowls), both with and without the exchange rule, respectively (For details see Wakatsuki et al. [16]). From these results, the average scores (with respect to the rankings) in the case of the 5-Skip rules are almost the same as those in the case of the UECda standard rules. Therefore, 5-Skip has no effect on the average scores and rankings.

Table 5: Average Scores Played by One Snowl and Four Defaults with the Exchange Rule

Program	$Score_{UECda}$	$Score_{5Skip}$	$\#5S_{early}$	$\#5S_{all}$
Snowl(0)	12878.02	13093.58	1108.66	1259.69
Default(1)	8269.99	8325.27	1423.38	1862.44
Default(2)	7986.04	7885.80	1375.68	1867.01
Default(3)	7962.82	7853.98	1288.60	1778.76
Default(4)	7903.13	7841.37	1288.96	1776.86

Table 6: Average Scores Played by One Snowl and Four Defaults without the Exchange Rule

Program	$Score_{UECda}$	$Score_{5Skip}$	$\#5S_{early}$	$\#5S_{all}$
Snowl(0)	11593.05	11488.30	1535.17	1864.70
Default(1)	8602.30	8548.61	1487.88	1799.35
Default(2)	8298.88	8376.14	1441.63	1782.48
Default(3)	8271.07	8294.01	1405.86	1740.26
Default(4)	8219.71	8292.94	1411.96	1740.35

3.5 Discussion

From the above results, 11-Back has no effect on increasing the average number of possible moves B . As a result, 11-Back could not increase the strategic complexity \sqrt{B}/D . To increase the strategic complexity, it is necessary to increase B . Further, because the average number of possible moves when the field is lead B_{lead} is much larger than B , one of the easiest ways to increase B is to increase the number of times when the field is lead. This is achieved by introducing a new rule that ends a round, such as 8-Enders or 3 of spades.

On the other hand, 11-Back has the effect of reducing the gap between the titles. By combining this with other localized rules, it can be expected to reduce the gap further.

The difference in the range and time of the effect (due to the rules) is thought to be the reason that the 5-Skip rules have less influence on the difference in scores (according to the seat order) than the 6-Reverse rules.

From all the experiments, we found that one localized rule had little influence on the average game length, average number of possible moves, strategic complexity, and the difference of scores for each seat order.

4 Differences in the Play Style Between Humans and Computers

4.1 Research Method

We designed and conducted a questionnaire and collected logs from computers to determine the difference between humans and computer programs. The questionnaire format used was the Web questionnaire by Google Forms; it was used to survey visitors to the UECda-2018 on November 24, 2018 and for visitors to our laboratories in the second open campus in UEC on November 25, 2018. A total of 71 valid responses were received.

The questionnaire consisted of 15 questions, which are those presented in the preliminary experiment and their variations. Participants were shown a player's initial hand with no card or one card on the table and were asked, "If you play Daihinmin with the UECda standard rules and this set of cards is your initial hand, which card(s) do you want to play or pass?"

In addition, for each Daihinmin program, each question was reproduced in a situation immediately after the game had started, and computer logs were taken 1,000 times for each question. We used the Snowl, Crow, Paoon, and Wisteria computer programs in this experiment.

4.2 Explanation of Questions

Among the 15 questions, some clearly show human features, which are shown in Figures 1–3. Cards on the right of the vertical line represent the initial hand. If a card is on the left of the vertical line, it is a previous card played at that time.

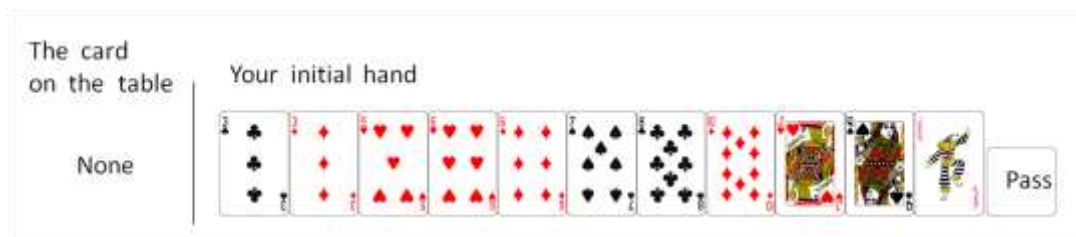


Figure 1: Question ID 0102

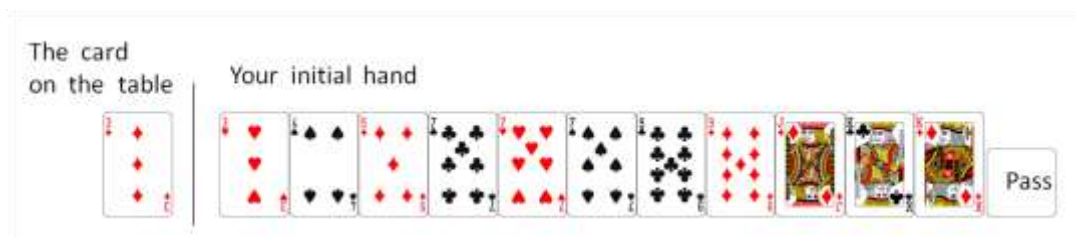


Figure 2: Question ID 0202

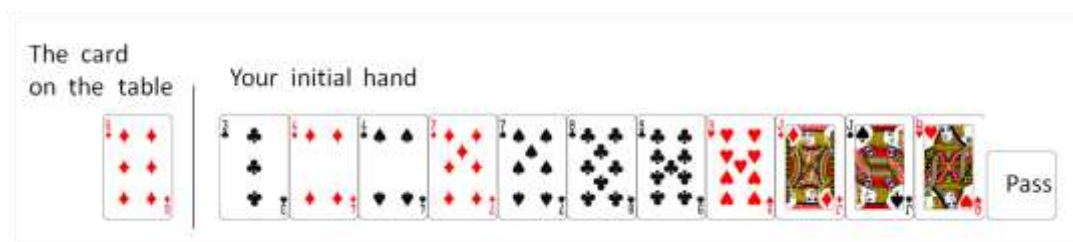


Figure 3: Question ID 0300

The question in Figure 1 examines whether or not a human plays the joker in their first turn. The question in Figure 2 examines the difference in submissions between humans and computer programs with the Lock. The question in Figure 3 examines the difference in submissions between humans and computer programs with the 8-Ender and the Lock.

4.3 Results for Questions

Results for the aforementioned questions are shown in Tables 7–9. Here, “Questionnaire” represents the ratio of the card(s) selected by humans. In addition, “Snowl,” “Crow,” “Paoon,” and “Wisteria” represent the ratio of the card(s) played by each program, respectively. “Others” is not possible in the standard UECda rules or no answer.

According to the results in Table 7, the submission of the joker in the first turn was very rare for humans, while it was not unusual for Snowl, Crow, and Wisteria to play the joker. In particular, the ratio of playing “Heart 5, Joker” or “Spade 7, Joker” exceeds 60% for both Snowl and Crow.

Table 7: Results for Question ID 0102 [%]

	Questionnaire	Snowl	Crow	Paoon	Wisteria
Club 3	5.6	0.0	0.0	0.1	0.0
Diamond 3	1.4	0.0	0.0	0.0	0.0
Heart 5	31.0	31.4	31.4	99.0	6.5
Diamond 6	2.8	0.0	0.0	0.0	0.0
Spade 7	2.8	3.3	1.1	0.1	0.0
Diamond 10	0.0	0.0	0.0	0.1	0.0
Heart J	0.0	0.0	0.0	0.1	0.0
Club 3, Diamond 3	46.5	0.0	0.0	0.3	91.0
Heart 5, Joker	0.0	59.7	64.9	0.0	0.0
Heart 6, Diamond 6	5.6	0.0	0.0	0.3	0.0
Spade 7, Joker	0.0	5.6	2.6	0.0	0.0
Heart 5, Heart 6, Joker	0.0	0.0	0.0	0.0	2.5
<i>Others</i>	4.2	---	---	---	---

Table 8: Results for Question ID 0202 [%]

	Questionnaire	Snowl	Crow	Paoon	Wisteria
Pass	1.4	0.0	0.0	0.0	0.0
Spade 4	45.1	97.1	99.9	81.4	100.0
Diamond 5	38.0	2.9	0.1	18.6	0.0
Club 7	1.4	0.0	0.0	0.0	0.0
Heart 7	1.4	0.0	0.0	0.0	0.0
Diamond 9	1.4	0.0	0.0	0.0	0.0
Diamond J	8.5	0.0	0.0	0.0	0.0
<i>Others</i>	2.8	---	---	---	---

From Table 8, the ratio of playing “Spade 4” is almost 100% for all computer programs. On the other hand, from the questionnaire result, the ratio that answered “Spade 4” is less than 50%, and the ratio that answered “Diamond 5,” which can be locked in this situation, is 38.0%. In addition, from Table 9, the ratio of humans that answered “Diamond 7” or “Diamond J” is 19.7%, while the ratio of playing “Diamond 7” or “Diamond J” by the computer programs is almost 0%. Therefore, we can see that humans use the Lock positively.

From Table 9, from the questionnaire results, the ratio of humans that answered “Club 8” is about 50%, while the ratio of playing “Club 8” is very low for all computer programs. Therefore, it is inferred that humans want to save pairs rather than an 8 in their first turn. On the other hand, the computer programs used in this experiment save an 8 in their first turn.

Table 9: Results for Question ID 0300 [%]

	Questionnaire	Snowl	Crow	Paoon	Wisteria
Pass	9.9	21.3	18.3	2.0	60.7
Diamond 7	16.9	0.0	0.0	0.0	2.1
Spade 7	4.2	0.0	0.0	0.0	3.5
Club 8	49.3	3.9	1.3	4.1	0.3
Heart 9	0.0	0.0	0.0	0.1	7.0
Club 9	2.8	0.0	0.0	0.0	6.7
Diamond J	2.8	0.2	0.0	0.0	7.0
Spade J	0.0	0.1	0.0	0.0	8.9
Heart Q	9.9	74.5	80.4	93.8	3.8
<i>Others</i>	4.2	---	---	---	---

5 Conclusion

We presented universal estimates, which are several measures of the complexity of the card game Daihinmin, to obtain several of its characteristics. We conducted computer experiments to obtain these estimates and investigated the effect of localized rules on Daihinmin computer programs. As a result, the following properties were clarified:

- (1) 11-Back has no effect on increasing the average number of possible moves and the strategic complexity.
- (2) 11-Back has the effect of reducing the gap between the titles.

- (3) 6-Reverse affects the difference in the score according to the seat order.
- (4) 5-Skip has less effect on the difference in the score according to the seat order.

Next, we showed the difference in the play style of Daihinmin between humans and computers using winning programs from the UECda. The major difference between humans and computers is that a human rarely plays a joker in the first turn, evaluates the value of the 8-End lower than the computers, and evaluates the value of the Lock higher than the computers. In future work, it is expected that a computer program will be developed that will feel human-like based on these results.

References

- [1] Daifugo – Wikipedia, [https://en.wikipedia.org/wiki/Daifug\C5\%8D](https://en.wikipedia.org/wiki/Daifug%C5%8D).
- [2] The UEC Computer Daihinmin Convention. <http://www.tnlab.inf.uec.ac.jp/daihinmin/>. (in Japanese)
- [3] M. Konishi, S. Okubo, T. Nishino, and M. Wakatsuki, “Decision tree analysis in game informatics,” *Studies in Computational Intelligence, “Applied Computing & Information Technology,”* Springer International Publishing, vol.727, 2018, pp. 13-27.
- [4] M. Konishi, S. Okubo, T. Nishino, and M. Wakatsuki, “A decision tree analysis of a multi-player card game with imperfect information,” *International Journal of Software Innovation*, vol. 6, issue 3, 2018, pp. 1-17.
- [5] M. Minsky, “The Society of Mind,” New York: Simon and Schuster. ISBN 0-671-65713-5, 1988.
- [6] K. Ohto and T. Tanaka, “Supervised learning of policy function based on policy gradients and application to Monte Carlo simulations in Daihinmin,” *IPSJ SIG Technical Report*, vol. 2016-GI-35, no. 10, 2016, pp. 1-8. (in Japanese)
- [7] S. Okubo, T. Ayabe, and T. Nishino, “Feature extraction and cluster analysis using n-gram statistics for DAIHINMIN programs,” *Studies in Computational Intelligence, “Applied Computing & Information Technology,”* Springer International Publishing, vol. 619, 2016, pp. 27-41.
- [8] S. Okubo, T. Ayabe, and T. Nishino, “Cluster analysis using n-gram statistics for Daihinmin programs and performance evaluations,” *International Journal of Software Innovation*, vol. 4, issue 2, 2016, pp. 33-57.
- [9] S. Okubo, Y. Kado, Y. Takeuchi, M. Wakatsuki, and T. Nishino, “Toward a statistical characterization of computer Daihinmin,” *International Journal of Software Innovation*, vol. 7, issue 1, 2019, pp. 63-79.
- [10] S. Okubo, M. Wakatsuki, T. Mitsuishi, Y. Dobashi, and T. Nishino, “New applications of the Monte Carlo tree search to computer Daihinmin,” *International Journal of Smart Computing and Artificial Intelligence*, vol. 4, no. 1, 2020, pp. 18-35.
- [11] F. Suto, K. Narisawa, and A. Shinohara, “Development of client “Snowl” for computer

- Daihinmin convention,” Computer Daihinmin Symposium 2010, 2010. (in Japanese)
- [12] Y. Tajima and K. Tagashira, “Heuristics for Daihinmin and their effectiveness,” *Studies in Computational Intelligence, “Applied Computing & Information Technology,”* Springer International Publishing, vol. 619, 2016, pp. 59-69.
- [13] Y. Tajima, K. Tagashira, and G. Kikui, “Heuristics for Daihinmin and their effectiveness,” *International Journal of Computer & Information Science*, vol. 17, no. 2, 2016, pp. 7-14.
- [14] M. Wakatsuki, M. Fujimura, and T. Nishino, “A decision making method based on Society of Mind Theory in multi-player imperfect information games,” *International Journal of Software Innovation*, vol. 4, issue 2, 2016, pp. 58-70. DOI:10.4018/IJSI.2016040104.
- [15] M. Wakatsuki, Y. Dobashi, T. Mitsuishi, S. Okubo, and T. Nishino, “Strengthening methods of computer Daihinmin programs,” *Proc. of 30th International Conference on Computer Applications in Industry and Engineering (CAINE 2017)*, San Diego, California, USA, 2017, pp. 229-236.
- [16] M. Wakatsuki, Y. Kado, Y. Takeuchi, S. Okubo, and T. Nishino, “What are the characteristics of the card game Daihinmin?,” *Proc. of the 8th International Congress on Advanced Applied Informatics (IIAI AAI 2019)*, 2019, pp. 587-592.