Heuristics for Daihinmin and their effectiveness

Yasuhiro Tajima[ab], Kouzou Tagashira[c], Genichiro Kikui[a]
Okayama Prefectural University

Abstract

Heuristics for games is one of the most important factors to solve or to make a game playing program. Especially, there are many heuristics for perfect information games. On the other hand, imperfect information game is not studied enough to make a program which can overcome a human player. In this paper, we show four heuristics for the card game "Daihinmin" and evaluate their effectiveness by some experiments. Our heuristics are implemented on the program called ``kou'' which is the champion program at UECda-2014 light class. This program uses only heuristics to play Daihinmin. In evaluation experiments, we show the strongness of ``kou'' by matching past champions which contain both of Monte-Carlo algorithm and heuristic algorithm. The result is that our heuristics are effective and our program is as strong as Monte-Carlo algorithm.

Keywords: Daihinmin, imperfect information game, heuristics, card game, evaluate function.

1. Introduction

Game study is one of the symbolic field of study of intelligence. If a computer wins a human, everybody would recognize a solid progress of computer science. Study on games can be classified into perfect information games and imperfect information games. The perfect information game can be solved by game tree search. The $\alpha - \beta$ method is a famous game tree searching method. In recent years, Monte-Carlo tree search[3] is known as effective method for game tree search. Especially, Go is the most successful application of Monte-Carlo tree search[4]. Computer Shogi is also the frontier of studies on perfect information games.

In [5], we can see that recent computer systems can win a professional shogi player in even conditions. Solving these perfect information games is usually executed by a deterministic algorithm. Then, these results can be useful for another algorithm study. Especially, automata theory and its application[10] is the basic and important problem of computer science. Game tree search algorithm and studies on perfect information games bring useful results for these fields.

For perfect information games, heuristics are also important as the game tree search algorithm. Even if there is a fast game tree search algorithm, a bad evaluation function leads lose easily. To make a good evaluation function, heuristics for the game is important. Recently, many evaluation functions are made by machine learning method but features of machine learning are usually constructed from heuristic ideas of the games.

Unfortunately, such heuristic ideas can not be re-used for another game. Heuristics are deeply dependent on each game and specific heuristics tend to be more important for the game.

On the other hand, there is no effective tree searching method for imperfect information games. There is a difficulty that possible states are too many because of imperfectness, and then there are too many branches if we create a game tree. But some studies and progress can be found on imperfect information games [1][2]. Many card games are imperfect information games. In addition, many card games uses standard pack of 52 cards with one or two jokers. Thus, we can regard these card games as imperfect information but not random game except the first shuffling.

"Daihinmin" is a famous card game in Japan. This game is similar to Big-Two which is also a card game mainly played in China[7]. The player is more than 2, and then Daihinmin is a multi-player, imperfect information game. There is a competition of computer Daihinmin called "UECda"[11]. In this annually competition, Monte-Carlo simulation method is an effective move searching method[8] and all champions uses this method in recent years.

In this paper, we show some heuristics for Daihinmin and evaluate whose effects by experiments. The base program to implement our heuristics is called "kou" which is the champion program at light class in UECda-2014 and this does not use Monte-Carlo method. This program is as strong as "snowl"[9] which is the champion program at UECda-2010 and uses Monte-Carlo method. In addition, the time complexity of our program is about one hundred

[a] Department of Systems Engineering,
111 Kuboki, Soja, Okayama, 719-1197 Japan
[b] tajima@cse.oka-pu.ac.jp
[c] Graduate School of Computer Science and Systems Engineering,
111 Kuboi, Soja, Okayama, 719-1197 Japan

times faster than that of Monte-Carlo method program. This is a new possibility of heuristics on computer Daihinmin.

2. Rules of "Daihinmin"

Daihinmin is a multi-player imperfect information game and uses standard pack of 52 cards with one joker. The number of players is arbitrarily more than 2 but 5 players are the best balance to play. In the following, we assume that the number of players is 5.

At first, shuffled almost even number of cards are provided to every player. The 53 cards can not be divided evenly by 5 players, thus three players have 11 cards and two players have 10 cards. The rank of cards has the order 3,4,5,6,7,8,9,T,J,Q,K,A,2 where "2" is the strongest card and "3" is the weakest. In contrast, the suit "c", "d", "h", "s" is not ordered.

The goal of this game is to get rid of all hands as fast as possible, and

- the fastest player is called "Daifugo",
- the second fastest player is called "Fugo",
- the third player is called "Heimin",
- the fourth player is called "Hinmin" and
- the fifth, that is the latest, player is called "Daihinmin."

The round ends when Daihinmin is decided. One round consists of some tricks and a trick is led by the player who has taken the previous trick. The first trick in the round is led by the player who has the card "3d." The player who leads a trick can play arbitrarily and the trick is followed by plays. Every player sits along a playing order. The order is changed after some rounds end.

The "play" of this game can be classified into followings.

- Every single card is a "Single" play.
- Plural cards are one of "Pair", "Triple" or "Quartet" play if all ranks of cards are the same.
  For example, a set of "4s" and "4d" is Pair and "Qc, Qh, Qs" is Triple.
  On the other hand, "7s, 8s" or "6c, 6d, 8h" are not a play.
- The play named "Kaidan" is constructed of more than three cards such that all cards have the same suit and ranks are successive.
  For example, "4h, 5h, 6h" is Kaidan with three cards, but both of "6c, 7s, 8c" and "8d, Td, Jd, Qd" are not Kaidan.

We call "Single", "Pair", "Triple", "Quartet" or "Kaidan" with the number of cards "types of play."

For example, the type of "7s" is Single, the type of "Ts Js Qs Ks" is Kaidan with four cards and the type of "3c, 4c, 5c" is Kaidan with three cards.

Every type of play has a specific order of strength as follows.

- On the type of Single, Pair, Triple or Quartet :
  The rank of the play represents the order.
  For example, on the type of Triple, "6s, 6c, 6h" is weaker than "7h, 7d, 7c" and it is stronger than "4h, 4d, 4s." On the other hand, "6d, 6c, 6h" has the equal strength.
  For another example, "2h, 2d" is the strongest on the type of Pair.
- On the type of Kaidan :
  Let α and β be plays whose type is Kaidan. If the highest rank in α is lower than the lowest rank in β then β is stronger than α, or vice versa. If the condition is not satisfied then α and β are incomparable.
  For example, "3d, 4d, 5d" is weaker than "6c, 7c, 8c" but "4s, 5s, 6s" is incomparable to both of them.
  Of course, "3d, 4d, 5d" and "6h, 7h, 8h, 9h" are not in the same type then these two plays are incomparable.

The player can select pass or play. Every play must be stronger than the last play in the trick. If the player has no stronger play then he must pass.

Once pass is selected, the player can not play until the trick has been ended. When all players pass, the trick is taken by whom the last play does.

From the shuffled card providing, this game is non-deterministic. This game is zero-sum because one player becomes Daifugo then other player can not be Daifugo. Thus, the game "Daihinmin" is multi-player, zero-sum, finite, non-deterministic, imperfect information game.

There are some special rules depend on play.

- Joker can be used as a substitution of any cards.
  The play "7d, 9d, Joker" is equivalent to "7d, 8d, 9d." Therefore, "Ac, 2c, Joker" is the strongest Kaidan with three cards. The single Joker is the strongest Single but only "3s" can defeat it.
- "Shibari" :
  When succeeding two plays have corresponding suits, then the trick is in Shibari.
  If Shibari is initiated, all succeeding plays must also have corresponding suits. In other words, when Shibari begins then possible plays are restricted such that the suits is corresponding until the end of the trick.
  For example, assume that the last play is "6d" and next player plays "9d", then the successive player can only play "Td", "Jd", "Qd", "Kd", "Ad", "2d" or "joker."

Assume that he plays "Ad" then the next player can only play "2d" or "joker."

For another example, assume that the last play on the trick is "7s, 7d" and the next player plays "Qs, Qd", then the successive player can only play "Ks, Kd", "As, Ad" or "2s, 2d."

- "8" (eight card) :

  If a play contains a card whose rank is "8", then the trick is terminated. Then a new trick is led by this player.
- "kakumei" (=revolution) :

  Quartet or Kaidan with more than five cards can initiate the "kakumei" (which means "revolution").

  When kakumei begins, the order of the rank becomes upside down. In the revolution, the rank "3" is the strongest and "2" is the weakest. Twice kakumei brings the order into normal.
- Card exchange :

  When a successive round starts, following card change is done after dealing the cards.
  - Daihinmin (5th at the last round) must submit the best and the second best Single card to Daifugo (1st at the last round), and Daifugo must return arbitrary two cards to Daihinmin.
  - Hinmin (4th at the last round) must submit the best Single card to Fugo (2nd at the last round), and Fugo must return arbitrary one card to Hinmin.

When a round has been ended, every player gets the following points:

- Daifugo : 5 points
- Fugo : 4 points
- Heimin : 3 points
- Hinmin : 2 points
- Daihinmin : 1 point

After plural rounds, players' ranking is decided by their total points.

3. UECda – The comuter Dihinmin competition

In the University of Electro-Communications, Japan, computer Daihinmin competition is held every year[11]. The competition has two classes. One is "light class" whose program is restricted to heuristic or some light algorithms. The other is "unlimited class" in which any algorithm is allowed. The competition sets the following environment.

- The number of player is just 5.
- After 100 rounds are played, players' playing order (= sitting position) is changed randomly.
- Total 1000 or 4000 rounds are done and the total of points decides the players' rank.

In UECda-2014, we won the "light class" with the program named "kou" and we introduce our heuristics in this paper.

4. Heuristics

4.1 Algorithm overview

The following is the overview of the algorithm of "kou" which is the winner of UECda-2014 light class.

1. Finding better combination of plays in the hand such that
   - each of plays has no overlapping and
   - the number of plays is nearly minimum.

2. Select the play which has the highest evaluation value to end the trick. We define the evaluation function for this selection. This function includes some heuristics and special rules to make a priority on plays.
3. If such play is not found, select the play such that "weakness of the hand" is not increase. This means that remaining hand is stronger than the present hand. Usually, the weakest will be played with this criterion.

The combination of plays in hand is made by the following algorithm.

1. Find all Kaidan whose cards are not any member of Pairs, Triples or Quartets.
2. Let the strongest card and "8" be Single.
3. Find Quartets, Triples and Pairs from the rest of the hand.
4. At last, all the rest cards are Singles.

This algorithm divides player's hand into plays which are not overlapping.

Heuristics used in the evaluation function are as follows.

4.2 Evaluation to end the trick

The evaluation value is found for each play in the hand. If this value is higher than that of other plays, we think the play is stronger and it tends to end the trick. The evaluation value for a play is calculated by the followings.

1. If the play can end the trick, i.e. there is no possible play which is stronger than the play, then the evaluation value is 100. If the move contains "8" then it is 101.

   Joker's value is 100 if "3s" is already played, otherwise it is 1.
2. If the play is Kaidan,

   100 - (possible Kaidan plays) * (finished player + 1)

is the evaluation value.

Here, (possible Kaidan plays) is the number of possible Kaidan which can be played by enemies.

(finished player) is the number of players who have finished this round.

3. If there are more than 3 players remaining and the play is Pair, Triple or Quartet then

$$100 - \sum_{i=1}^{m} f_i(n)$$

is the evaluation value.

Here, m is the number of possible stronger plays which is in enemies' hand.

Each i means a play which is in enemies' hand.

$f_i(n)$ is found by the following.

Let n = (the number of stronger cards) - (the number of cards of my play) + (finished player).

Then, it is defined as follows.

$$f_i(n) = \begin{cases} 4 & (n \le 0) \\ 9 & (n = 1) \\ 15 & (n = 2) \\ 24 & (n \ge 3) \end{cases}$$

This means stronger play i takes smaller value of $f_i()$, then the evaluation value will be bigger.

4. Otherwise, i.e. if the move is Single, Pair, Triple or Quartet,

100 - ((stronger total) * 30)

is the evaluation value.

Here, (stronger total) is the sum of ranks of plays which are in enemies' hands.

5. If these values are less than or equal to 0, let it 1.

If this value of the play is equal to or greater than 95, then we think the play will end the trick. We call this value "strength" of the play.

Let m be a play, then the strength of m is denoted by s_m.

4.3 "Shibari" priority

When the player can initiate Shibari by the play m, the strength of such play s_m is modified to upper limit if the following condition holds.

- After this Shibari, the player has the strongest move.

If this condition does not hold, m will not be played.

4.4 Trick leading play

We define play priority for every play m to lead the trick.

The play priority p_m for a play m is calculated by the following.

- If m is Kaidan and contains "8" then $p_m = 9$.
- If m is Joker then $p_m = 3$.
- If $1 \le s_m \le 94$ then

  $p_m = 20 + 2(rank\ of\ m)$.

  Here, (rank of m) is 1 if play "m" is the weakest and 13 if play "m" is the strongest among plays whose types are the same.

  For example, a Kaidan "4s 5s 6s" has the (rank of m) of 2 because there exists only weaker play "3s 4s 5s."
- If $95 \le s_m$ then $p_m = 110 - s_m$.

If p_m is bigger, the play "m" has high priority. If there are some plays whose priority are the same, then the best play is selected from the following order :

Kaidan, Triple, Pair and Single.

This value is also used to play when there is no play to end the trick. In such case, combining the following hand weakness, the best play will be selected.

4.5 Strong play reservation

We define the hand weakness. The hand weakness is the sum of the following w_m for every play m in the hand.

$$w_m = \begin{cases} 2 & (1 \le s_m \le 30) \\ 1 & (31 \le s_m \le 60) \\ 0 & (61 \le s_m \le 90) \\ -1 & (91 \le s_m) \end{cases}$$

This value means how the remaining hand is weak. If there is the play m such that s_m is high but the hand weakness is also high, then m will not be played. This is balancing procedure to avoid wasting strong moves.

5. Evaluation experiments

5.1 Comparing with past champions

The following is the list of past light class champions.

- "chibiHana" is the C version of Kishimen_2013 which is the champion of the light class in UECda-2013.
  This program makes combination of plays whose number is the minimum. The finish search of this program is sophisticated and Shibari strategy is also implemented.
- "Party" is the champion of the light class in UECda-2012.
  This program tends to have plays which can end the trick. Joker will be used to play weaker cards. Shibari is also considered to end the trick.

Comparing with these two programs, we set the match among the following five programs.

- kou
- chibiHana
- Party
- default
- default

Here, default is the basic strategy program which plays the weakest in the hand. Table 1 is the scores of this setting.

Table 1 : Results among light class champions

rounds	1000	4000	7000	10000
kou	3660	14954	26178	37413
chibiHana	3493	14236	24817	35460
Party	3271	12698	22208	31536
default	2324	9198	16017	22900
default	2252	9014	15780	22691

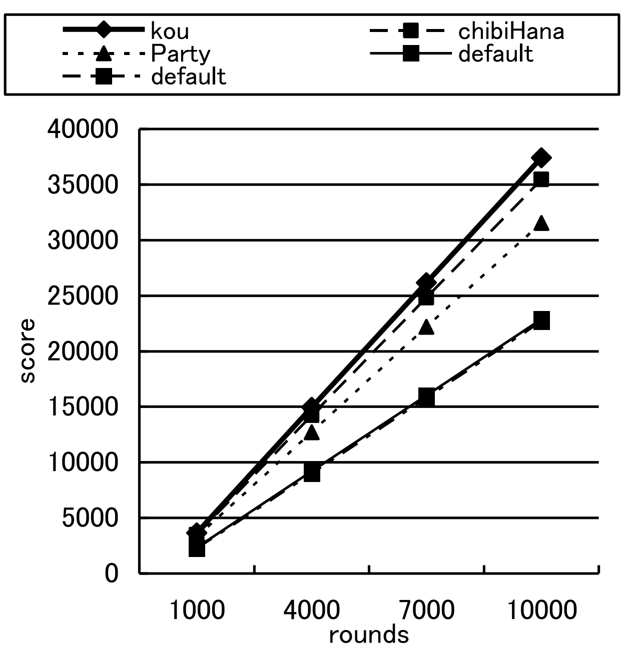

Figure 1 : Scores among light class champions

From this result, "kou" is the strongest heuristic algorithm comparing with past two years champions. Especially, "kou" marks high scores at all rounds from 1000 to 10000.

5.2 Comparing with Monte-Carlo algorithms

Monte-Carlo algorithm is also effective to Daihinmin. In recent years, all champions at "unlimited class" are made with Monte-Carlo simulation. In Daihinmin, we can not make a search tree because of imperfectness. Thus, all programs have the depth 1 tree for Monte-Carlo simulations. In addition, opponents' hands are simulated by random card delivering. The following is the list of famous Monte-Carlo Daihinmin programs.

- "beersong" is the champion program at UECda-2013.
- "paoon" is the champion program at UECda-2012.
- "crow" is the champion program at UECda-2011[6].
- "snowl" is the champion program at UECda-2010[9].

Table 2 is the scores of these four programs and "kou."

Table 2 : Results against Monte-Carlo programs

rounds	100	300	500	700	900
kou	344	920	1481	2021	2528
beersong	335	999	1541	2213	2853
paoon	299	891	1570	2209	2893
crow	246	851	1451	2068	2678
snowl	276	839	1457	1989	2548

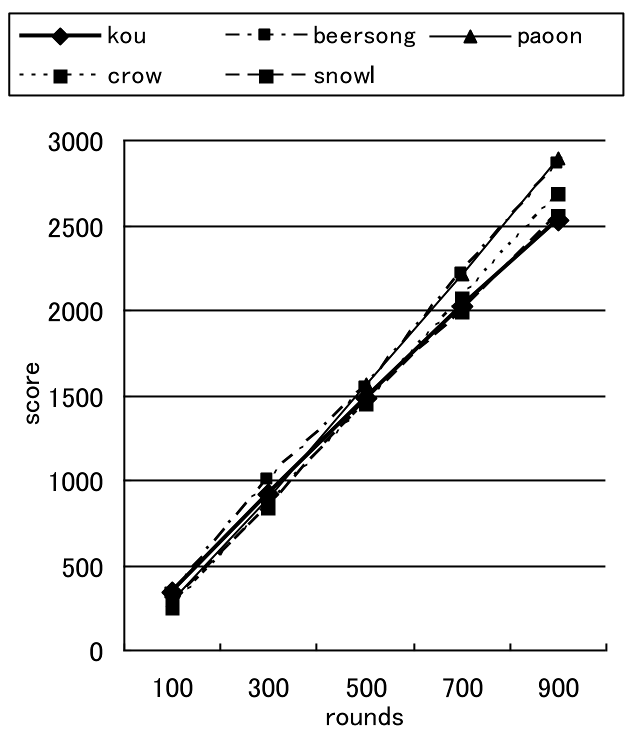

Figure 2 : Scores against Monte-Carlo programs

From this result, “kou” is as strong as “snowl” which is the early Monte-Carlo algorithm client of Daihinmin.

Time complexity of “kou” is very low comparing with any Monte-Carlo programs. We don't have precise data but kou's calculation time is about hundred times faster than other Monte-Carlo programs. Thus, we can say that kou marks the highest scores per second.

5.3 Heuristics effect

To evaluate the heuristics used in “kou”, we have made some modified players to compare with the original “kou.”

The modified player is as follows.

1. Evaluation to end the trick :
   The original “kou” uses 95 as the threshold value to decide that the play can end the trick or not. So, we prepare
   - w100,
   - w90 and
   - w80

   whose thresholds are 100, 90 and 80, respectively.

2. Shibari priority :
   We prepare programs
   - lock+ and
   - lock-.

   Here, lock+ plays Shibari if it possible and lock- never plays Shibari.

3. Trick leading play :
   Two programs are prepared which are
   - weak and
   - single.

   At the leading of the trick, weak always plays the weakest play in the hand. On the other hand, single always plays the weakest Single.

4. Strong play reservation :
   The program
   - use_2 : which does not use “hand weakness” value and “Strong play reservation” heuristic.

Every modified program and the original “kou” are independently matched with the four same enemies. Here, all enemies are the same program which is one of chibiHana, Party or default. For example, we set a test match with one w100 and four chibiHana.

In the followings, all experiments are evaluated by 1000 rounds.

5.3.1 Evaluation to end the trick

Table 3 shows the score difference between the modified program and the original program.

Table 3 : Score difference of end the trick heuristic

modified program	w100	w90	w80
opponent	score diff.	score diff.	score diff.
chibiHana	-2132	+403	+210
Party	-767	+663	+970
default	-825	+15	+121

The program w100 is weaker than the original from this result. On the other hand, w90 and w80 are stronger than the original, thus there will be more optimal threshold value under 95. For chibiHana, the threshold may be between 90 and 95, but it will be under 80 for Party and default. More detailed experiments and analysis are needed to decide the threshold value. Now, we can conclude that 95 is not bad value and this heuristic is effective for various enemies.

5.3.2 Shibari priority

Table 4 shows the score difference between lock+ or lock- and the original program.

Table 4 : Score difference of Shibari heuristic

modified program	lock-	lock+
opponent	score diff.	score diff.
chibiHana	-1211	-720
Party	-117	-287
default	-757	-203

From this result, the score of lock- is 1211 less than that of the original. Every difference is minus then we can conclude that both of lock+ and lock- are weaker than the original “kou.” Especially, the difference against chibiHana is bigger than the others. This is caused that chibiHana's “end the trick” strategy is more effective than Party and default.

5.3.3 Trick leading play

Table 5 shows the score difference about weak and single.

Table 5 : Score difference of trick leading play heuristic

modified program	weak	single
opponent	score diff.	score diff.
chibiHana	-165	-1086
Party	-679	-376
default	-856	-895

Both modified program weak and single become weaker than the original.

Thus we can say that the trick leading heuristic is also effective. For chibiHana, single’s score is lower than over

1000 points. We can say that types of plural cards are more important against chibiHana. On the other hand, weak's score is lower than single against Party. It implies that selection of the trick leading play is effective against Party. There are not difference between the score of weak and single against default. This is from that default also leads the trick by single and the weakest play.

5.3.4 Strong play reservation

Table 6 shows the score difference about use_2.

Table 6 : Score difference of strong play reservation heuristic

modified program opponent	use_2 score diff.
chibiHana	-157
Party	+100
default	+102

Evaluation of this heuristic's effectiveness needs careful discussion. For chibiHana, this heuristic is effective. But it is not effective for the other two enemies. Obviously, default is the weakest among chibiHana, Party and default. Thus, strong play reserving is effective for the enemies which has complicated algorithm. For a simple enemy, simple play will be effective rather than reserving strong plays.

6. Conclusions

In this paper, we introduce the heuristics of "kou" which is the champion program at UECda-2014 light class. The main heuristics are the following four point of view.

- Evaluation to end the trick
- Shibari priority
- Trick leading play
- Strong play reservation

All of these heuristics are effective but Strong play reservation is not effective for some simple algorithm enemies. Moreover, there is possibility that the threshold value or some parameters can be sophisticated by optimization technique or machine learning method.

Now, such parameters are provided by hard coding in the program and the values are also selected by heuristic method. When we optimize these values then our program will become more stronger. This is remained for a future study.

In addition, we have shown the strongness of "kou" against past champions and Monte-Carlo algorithms.

Especially, "kou" is little stronger than "snow" which is the champion program at UECda-2010 and it is famous as the turning point that Monte-Carlo algorithm is useful for Daihinmin.

It is known that Monte-Carlo algorithm takes much time to play. By our brief counting, heuristic algorithms are one hundred times faster than Monte-Carlo algorithms. Thus, We can show a new possibility of Daihinmin player by heuristic algorithm.

References

[1] E. Bonnet, F. Jamain, A. Saffidine, On the Complexity of Trick-Taking Card Games, Proc. of 23rd IJCAI, pp.482--488, 2013.

[2] M. Bowling, N. Burch, M. Johanson, O. Tammelin, Heads-up Limit Hold'em Poker is Solved, Science, Vol.347, No.6218, pp.145--149, 2015.

[3] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis and S. Colton, A Survey of Monte Carlo Tree Search Methods, IEEE Trans. on Computational Intelligence and AI in Games, Vol.4, No.1, pp.1--49, 2012.

[4] R. Coulom, Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search, Proc. 5th International Conference on Computer and Games, pp.72--83, 2006.

[5] S. Konuma, T. Honda, K. Hoki, T. Nishino, Application of Temporal Difference Learning to Computer Daihinmin, IPSJ SIG Technical Report, Vol.2012-GI-27, No.1, pp.1--4, 2012. (in Japanese)

[7] K. Hoki, T. Kaneko, D. Yokoyama, T. Obata, H. Yamashita, Y. Tsuruoka, and T. Ito, Distributed-Shogi-System Akara 2010 and its Demonstration, International Journal of Computer & Information Sciences, Vol.14, No.2, pp.55-63, 2013.

[8] J. Nishino and T. Nishino, Parallel Monte Carlo Search for Imperfect Information Game Daihinmin, Proc. of 5th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp.3--6, 2012.

[9] Y. Sutou, K. Narusawa, A. Shinohara, Development of Daihinmin Client ``snowl", 2nd Symposium on UEC Daihinmin Competition, 2010. (in Japanese)

[10] M. Wakatsuki, E. Tomita, T. Nishino, A Polynomial-Time Algorithm for Checking the Equivalence for Real-Time Deterministic Restricted One-Counter Transducers Which Accept by Final State, International Journal of Computer & Information Science, Vol. 14, No. 2, pp.45-54, 2013.

[11] Daihinmin competition UECda-2014 : http://uecda.nishino-lab.jp/2014/

Yasuhiro Tajima received his Ph.D. from The University of Electro-Communications in 2001. He had joined Ishikawajima-Harima Heavy Industries Co., Ltd., and Tokyo University of Agriculture and Technology. Currently, he is an associate professor of Okayama Prefectural University. His research interests are in Machine Learning and Text Mining. He is a member of IEICE, JSAI and IPSJ..

Kouzou Tagashira has received his B.E. degree from Okayama Prefectural University in 2015. Currently he is a student of Graduate School of Computer Science and Systems Engineering, Okayama Prefectural University. He is the main programmer of the Daihinmin program called "Kou" and won the championship in light class at UECda-2014.

Genichiro Kikui his Ph.D. degree in informatics from Kyoto University in 2007, respectively. He joined the Nippon Telegraph and Telephone Corporation (NTT) in 1986, and he was a visiting researcher at the Center for the Study of Language and Information (CSLI) at Stanford University in 1997. He has been a professor with Okayama Prefectural University in Okayama, Japan since 2011. He is interested in natural language understanding.