



# Distinguishing between humans and computers in a card game with imperfect information

Seiya Okubo<sup>1</sup>, Masato Konishi<sup>2</sup>, Mitsuo Wakatsuki<sup>2</sup>, and Tetsuro Nishino<sup>2</sup>

<sup>1</sup> University of Shizuoka, Japan

<sup>2</sup> The University of Electro-Communications, Japan

## Abstract

The purpose of this study is to analyze the behavior of humans and computers, and distinguish, from the results of the analysis, between humans and computer programs. In this study, we focus on the players of Daihinmin, which is a card game with imperfect information. Although not many Daihinmin play logs for human players have been collected, research on the principles of human behavior has been successfully conducted. For that purpose, we propose a method to distinguish between human and computer players using decision trees to represent player characteristics and to compare the differences between them. To evaluate the effectiveness of the proposed method, we apply it to distinguish between human and computer players of Daihinmin. Based on the results, we examine the validity of the proposed method and discuss its potential.

## 1 Introduction

In recent years, artificial intelligence (AI) technology has rapidly developed and is expected to be applied in many fields, including gaming. Several studies have been conducted on computer and board games. For example, Alpha GO developed by Google has realized a program with strength exceeding that of humans. As such, in the field of games, programs with strength beyond that of humans have been realized. Of course, because these programs are stronger than humans, they do not act like humans.

On the other hand, some programs act just like humans. Examples of such technologies include chatbots that conduct chats and NPCs (Non-Player Characters) that act like players in a game. These technologies allow programs to interact and act like humans, so that a variety of services can run without human intervention. Thus, programs that behave more like humans are one of the main targets in artificial intelligence research. In this study, such computer programs are called bots.

Several studies attempting to distinguish between humans and bots have been conducted, and one distinction for a chatbot is passing the widely studied Turing test. In MMORPG (Massively Multiplayer Online Role-Playing Game), bots have successfully obtained large quantities of experience values and items. In such games, bots are prohibited because they present additional problems, such as real-money trading. Therefore, various methods are used for bot detection in a variety of fields and situations. In these studies, it is possible to

identify a bot, but difficult to explain the principle of action. In order to check the accuracy of our method, the principles of human action should be known.

In this study, we will focus on the card game DAIHINMIN. DAIHINMIN is a multiplayer imperfect information game. In game informatics, games are classified into perfect information games and imperfect information games. In an imperfect information game, some information is not disclosed to the players (e.g. mahjong and many card games); because of the hidden information, only uncertain simulations can be conducted, predicting the withheld information. This diversity of withheld information leads to the difficulty of game analysis.

The UEC Computer Daihinmin Convention is held at the University of Electro-Communications, Tokyo every year, bringing together competitive client programs that correspond to Daihinmin players and contesting their strengths. Also, many program logs can be used in research. However, there are few human logs available for research. Therefore, it is not always possible to directly apply analysis methods in which the number of logs is important for classifying humans and computer programs.

In this study, in order to classify and describe the actions of bots and humans, we propose a method using decision trees. In particular, our method focuses on the structure of the tree. Through computer experiments, we can verify the effectiveness of the proposed method by applying it to Daihinmin game logs.

## 2 Definitions

### 2.1 Decision Tree Analysis

Decision trees are often used for prediction and classification, as well as extracting the rules of analysis source data [3]. A decision tree is a data mining technique that generates a conceptual representation of a tree structure in which each internal node represents an explanation for classification, and each leaf represents a classification result. There are two types of decision trees: classification trees and regression trees. The classification tree is used to classify the type (object variable) from the value of the dependent variable, and the regression tree is used to predict the value of a function (object variable) from the value of the dependent variable. The variables that are the basis of variation are called dependent variables, and variables to be classified are called object variables.

### 2.2 Computer Daihinmin

Daihinmin is a card game mainly played in Japan, but similar games are played around the world. It is a multiplayer game of imperfect information that has been extensively studied in recent years [2, 5, 6, 7, 8, 9]. Computer Daihinmin refers to playing the game on a computer. The UEC Computer Daihinmin Convention (UECda) is an annual competition for computer Daihinmin programs [1]. In this study, we adopt the framework utilized by the UECda [1]. The details of the rules of UECda are shown in Appendix A.

It is also desirable for the UECda committees to detect bots in an accountable manner. In this type of competition, it is only necessary to be able to detect bots before the awards are given. The bot detection is not required in real-time.

A study was conducted on clustering programs from computer program logs [6]. The clustering results from this study reflect the history of the program's development (e.g. developer, parent-child relationship of the program). In another study, the characteristics of the behavior of each program are analyzed using decision trees [2]. In the previous study [2], an analysis

of the characteristic behaviors of Computer Daihinmin strategies are conducted for computer programs. There is also an analysis of human player's play styles [7]. In the previous study [7], the principles of human behavior are explored by collecting questionnaires from human players. However, in these studies, the characteristics of the program and that of humans have been analyzed discretely. In addition, no analysis of human player logs for thousands of games has been conducted.

In the future, mechanical classification is expected to be used in DAIHINMIN. In the previous study [6] mentioned earlier, the clustering of programs by features is realized. This result suggests that if the number of logs is sufficient, classification by machine is possible. On the other hand, if the condition (opponent, etc.) is changed, the clustering fails. Therefore, it is necessary to study features that are more valid and analyze why such features are valid. Research using decision tree analysis has also been conducted in DAIHINMIN to explain the characteristics [2]. The number of features in [2] is also small. However, it is easy to gain knowledge about important features since the visualization is based on decision tree analysis.

There are also factors in the collection of logs. In the study of DAIHINMIN, the play log of the program can be collected through computer experiments. On the other hand, there are few studies that collect logs of human. One study that collected logs is TohoDAIHUGO [4], which can be used. However, the number of logs is not enough to conduct a large-scale experiment.

Thus, there may be a situation where the features to be used for classification by machine learning are not clear, and there are not enough logs for analysis. Therefore, it is necessary to have a method that can classify or provide clues for classification in such situations and a method to obtain knowledge of feature values that will be necessary for future classification by machine learning.

### 3 Proposed Method

If a large number of human logs are collected, and if there are uniform characteristics of human behavior, machine learning methods, such as decision forest, may be able to distinguish between programs and humans. However, if the collection of human logs is small, such a method cannot be used. On the other hand, in the field of games, research on human psychology and the characteristics of human behavior has been conducted[7]. Some of these research methods can be implemented using a small number of logs. Therefore, we propose a method to distinguish between humans and programs by using decision trees to represent the behavior characteristics and subsequently analyzing the trees. The proposed method presents behavior as a classification tree of the decision tree and compares the features of the obtained tree. In doing so, we read the program features and human psychological behaviors from the trees. Thus, the distinction can be achieved even in instances where there is no large-scale human logging.

The proposed method consists of three steps: collection of target logs, generation of decision trees, and evaluation of decision trees. The procedure is as follows:

Phase 1: Prepare Data

1. Collection of logs
  - (a) In the target environment, consider characteristic behaviors.
    - Each player plays the game while communicating with DAIHINMIN Server.
  - (b) Collect the behavior logs of the target player.
    - The DAIHINMIN server stores a log that can reproduce the game. For example, it stores the seating order, combination of players, hand information, and information on the cards played in each turn.

## Phase 2: Evaluate the Decision Trees

### 1. Generating decision trees

- (a) From the logs, extract data corresponding to dependent variables and object variables.
  - Determine the players to be analyzed.
  - Retrieve the logs of the target players and representative programs from the database.
  - Determine each variable based on previous research and findings (Special cards, number of turns, etc.) used in DAIHINMIN.
- (b) Using the data, generate decision trees.
  - Use a program such as R to generate a decision tree.

### 2. Evaluation of decision trees

- (a) Analyze the available features in the decision tree.
  - Compare the generated tree and each parameter. For example, the types of nodes that come to the top and the variables that are important.
- (b) From the features, determine whether the target is a human or a bot.
  - Determine whether a player is a suspicious player based on whether they have the same characteristics as the programs.

In phase 2-2, we determine whether the target is a human or bot by comparison. There are several methods of analyzing trees. The following are some examples of some of the methods. The first is to evaluate the features of the trees by comparing decision trees. By comparing the obtained features, we analyze the features of programs and humans. Based on the results, we can classify bots and humans. The second method is to compare the results with the human mind or the results of a questionnaire. Though people do not always behave in a rational manner they may behave in a mind-dependent manner. We can classify bots and humans by checking whether those behaviors appear in the decision tree.

## 4 Computer Experiments

### 4.1 Outline

We verified the validity of the proposed method using computer experiments.

We conducted the following two computer experiments. The first, Computer Experiment 1, clarifies the relationship between the decision tree analysis results and the number of logs. The logs for each human player are not collected in large quantities, thus the analysis is conducted on a small number of logs. In order to use the proposed method, the number of logs that work effectively must be known. Therefore, we evaluated the impact of the number of logs on the proposed method. In the second experiment, Computer Experiment 2, we analyzed the logs of human and computer programs, respectively, comparing the results to reveal differences. We also discuss how human behavior is based on psychological factors, and the differences between computer programs and humans. Based on the results of these experiments, we discuss the distinguishability of programs and human.

**Table 1:** Values

Variables	Data Type	Range	Description in Figs.	Value Type
Number of turns	int	0-	NumTurn	dependent
Total number of all players' cards	int	0-53	NumAllCards	dependent
Number of target cards	int	0-11	NumOwnCards	dependent
Have a Joker	Bool	0,1	HasJoker	dependent
Have a Group	Bool	0,1	HasGroup	dependent
Have a Sequence	Bool	0,1	HasSeq	dependent
Type of published cards	strings	single, group, sequence	N, G, S	object

## 4.2 Computer Experiment 1

The quantity and quality of logs are important during analysis. Particularly for human targets, it is difficult to collect large-scale logs. However, analysis may be performed from relatively few logs. In order to verify how the proposed method is affected by the number of logs, we applied it to Daihinmin. The program to be analyzed were Snowl. The three program chosen were used in UECda: Snowl [8] uses the Monte Carlo method.

The dependent values and target values are shown in Table 1. These values indicate the type of cards submitted at lead. In Daihinmin, when the field is lead (there are no cards on the table), submitted cards are not affected by the cards on the table. The program can submit cards freely and the features describing the type of submission appears.

We collected the logs from 500 games, 1000 games, and 5000 games. Each decision tree is shown in Fig. 1. The nodes of the decision tree contain the variable to be judged. The edges contain the branching by variable value. The leaves contain the type of play and each frequency. For example, the root node in Fig. 1 indicates that the branch is to the left if the player has a joker, and to the right if the player does not. The leftmost leaf shows that if a player has a joker and has two or fewer cards, the player has a high probability of playing a “Group.”

The structure of the six nodes in the upper portion of the tree remains the same for 1000-5000 games. However, there are differences at a tree depth of three or more; this may be due to pruning of the decision tree by rpart.

In this experiment, the complexity of the tree increased as the log size increased. However, logs of more than 1000 games produced a stable tree. As mentioned above, even if the number of logs change, the primary part does not; thus, it is considered possible to read the characteristic.

## 4.3 Computer Experiment 2

In order to determine whether the proposed method can identify humans and programs, we applied the proposed method to the logs of three programs and two humans. The three programs chosen were used in UECda: Snowl [8] and Wisteria [5] use the Monte Carlo method, and Kou2 [9] is a program that uses the evaluation value method, and is also based on the human mind. They are ranked in descending order: Wisteria, Snowl, and Kou2. Logs were collected from 1000 games using the same program. Therefore, the number of games is 5000 for each program, and the number of play logs is 15000 on average.

For human player logs, we chose two players who have played many games of TohoDaifugo.



Figure 1: Decision trees for Snowl

In this application, a human player plays against four default programs, and play logs are collected. This is different from experiments on programs. Therefore, the analysis is conducted keeping in mind that the log collection methods are different. In addition, we used the logs of the two TohoDaifugo players who played the greatest number of games. The details of the human players are shown in Table 2. The strength of each program is also listed for reference.

The generated decision trees are shown in Figs. 2-5. One common feature of all players is that the root node is HasGroup. However, there are many differences at the other nodes.

From these characteristics, we can see the following.

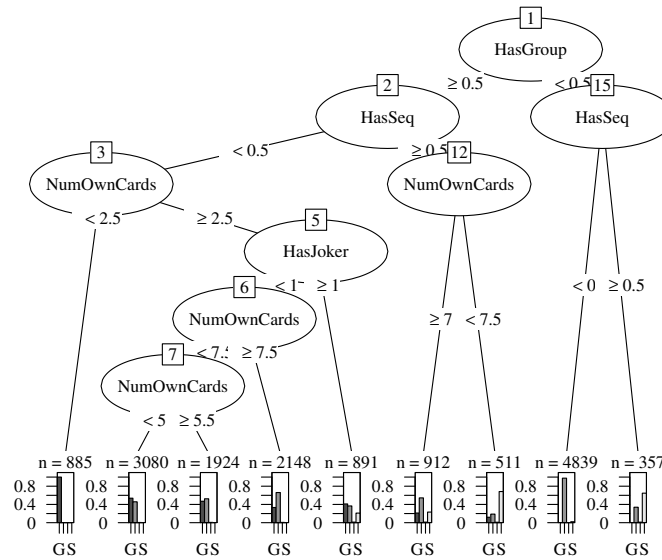
For the human players, NumTurn and NumOwnCards appear in the upper levels of the hierarchy. This means that human players are more aware of the number of cards in their hand and the progress of the game.

Programs evaluate “HasSeq” when “HasGroup” is false. In contrast, in the decision tree of human players, if there is no group, they either submit a single, or try to submit a group or sequence with a joker. We observed that humans tend to submit large numbers of cards with high priority to reduce their hands early and reduce the psychological burden. Therefore, it is

**Table 2:** Details of human players and the strength of each player

Player's name	Player ID	Number of games	Average of points
Human A	8c240f61333a890e585e0b8794ed83a352e695fe	5231	4.0258
Human B	472289394d3ec80bd172b4769294885e81f740ca	5492	3.6431
Wisteria			4.6369
Snowl			4.4039
kou2			4.2497

\* Player ID means "User ID of TohoDaifugo [4]"

**Figure 2:** Decision tree for Kou2

expected that a human player preferentially submits in the order of sequence, group, single.

The goal of Daihinmin is to submit all player's cards. For this reason, people psychologically prefer to play a large number of cards. On the other hand, the program only considers whether or not it will lead to victory; therefore, it has no concept of playing a large number of cards.

It is presumed that the program relies on rationality, while human psychology influences game play. The experimental results suggest that the decision trees bear this out. In the future, it is expected that it will be possible to further discriminate between programs and humans. Therefore, it is possible that classification could be achieved if finer player analysis is performed.

## 5 Conclusion

In this study, we proposed a method using a decision tree to distinguish between humans and bots in a gaming environment with human and non-human players. The proposed method generates decision trees from the collected play logs, reads features from the decision trees, and compares them. Even if the number of players is small, it is possible to distinguish between hu-

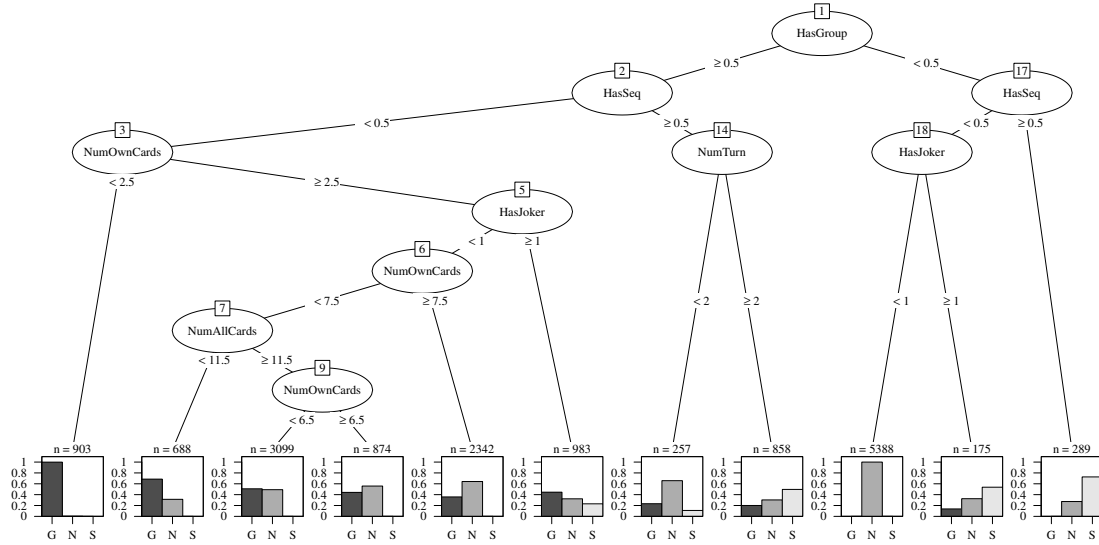


Figure 3: Decision tree for Wisteria

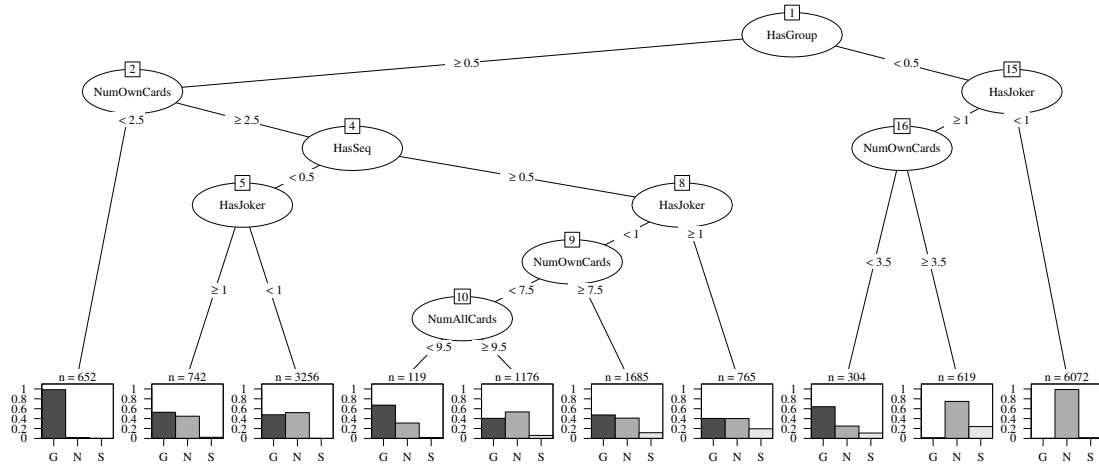


Figure 4: Decision tree for Human A

mans and programs by taking into account human psychology and other factors. Subsequently, to evaluate the effectiveness of the proposed method, we performed two computer experiments. The first provided an evaluation of the method’s robustness against the number of collected logs, and the results confirmed a certain degree of robustness. In the second experiment, we generated and compared the decision trees of four programs and two humans, examining their features. As a result, we found several defining features for each player. It became clear that it was possible to distinguish between humans and programs by reading this feature. The validity of the method in the context of Daihinmin was observed. If the principles of human behavior are known, the proposed method can be applied to other computer games.



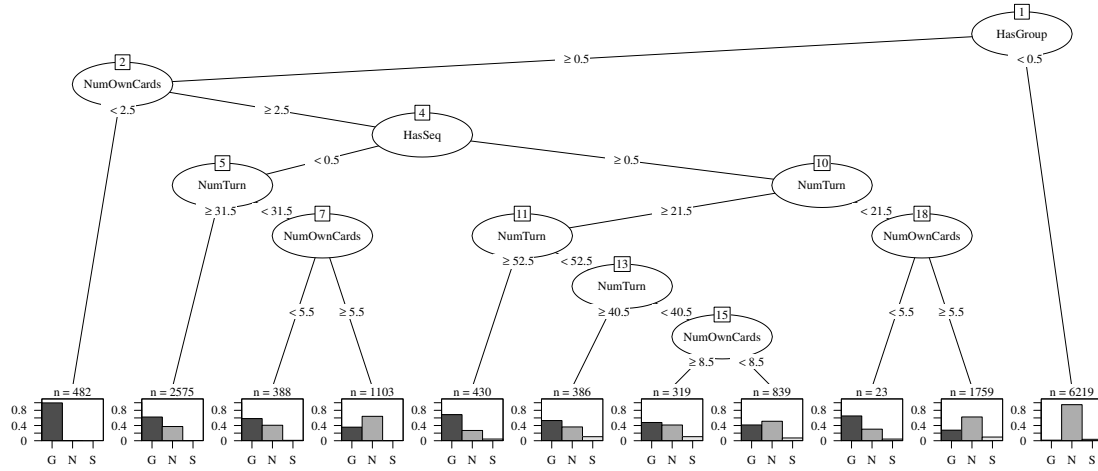


Figure 5: Decision tree for Human B

In the future, it is expected that features that enable more detailed analysis will be constructed based on the findings in this study.

## References

- [1] The UEC computer daihinmin convention. <http://www.tnlab.inf.uec.ac.jp/daihinmin/>.
- [2] Masato KONISHI, Seiya OKUBO, Tetsuro NISHINO, and Mitsuo WAKATSUKI. A decision tree analysis of a multi-player card game with imperfect information. *Int. J. Softw. Innov.*, 6(3):1–17, July 2018.
- [3] Tim MENZIES and Ying HU. Data mining for very busy people. *Computer*, 36(11):22–29, November 2003.
- [4] Masaki MITSUYAMA, Shinichi NABETA, Nobuaki MUTOH, Hiroaki YUZE, and Seiya OKUBO. Development of log collection system of daihinmin using android terminal. *The 7th National Convention of IPSJ*, 2013(1):107–108, Mar 2013.
- [5] Katsuki OHTO and Tetsuro TANAKA. Supervised learning of policy function based on policy gradients and application to monte carlo simulation in daihinmin. *IPSJ SIG Technical Report*, 2016-GI-35(10):1–8, Mar 2016.
- [6] Seiya OKUBO, Takaaki AYABE, and Tetsuro NISHINO. Cluster analysis using n-gram statistics for daihinmin programs and performance evaluations. *International Journal of Software Innovation (IJSI)*, 4, Issue 2:33–57, 2016.
- [7] Seiya OKUBO, Yuuta KADO, Yamato TAKEUCHI, Mitsuo WAKATSUKI, and Tetsuro NISHINO. Toward a statistical characterization of computer daihinmin. *International Journal of Software Innovation*, 7(1):63–79, 2019.
- [8] Fumiya SUTO, Kazuyuki NARISAWA, and Ayumi SHINOHARA. Development of client “snowl” for computer daihinmin convention. *Computer DAIHINMIN Symposium 2010*, 2010.
- [9] Kozou TAGASHIRA and Yasuhiro TAJIMA. Heuristics implementation and evaluations for computer daihinmin. *IPSJ Journal*, 57(11):2403–2413, Nov 2016.

## A Rules of DAIHINMIN

The rules used in The UEC Computer Daihinmin Convention are as follows:

### Game Procedure:

The game is played by five players and uses a total of 53 cards: 13 each hearts, clubs, spades, and diamonds, as well as a joker. The cards are ranked in the order: 3, 4, 5,  $\dots$ , 10, jack, queen, king, ace, 2, where 3 has the lowest value and 2 has the highest value. At the start of each game, each player is dealt 10 or 11 cards. Players take turns in clockwise order to discard (play) the cards in their hands. The first player to get rid of all their cards is the winner.

### Start of the Game:

The game starts with the player who has the 3 of diamonds. The player either plays (discards) their card(s) or passes the turn. This process is repeated for each player. If there are no cards on the table (“the field is lead”), a player may play any type of card in turn (a single, group [pair], or kaidan [sequence]). In other words, players can play any type and any number of cards. If a previous play is on the table (“the field is follow”), the current player can play their card(s) to defeat the previous play. Meaning, players need to play the same type, the same number of cards, and be stronger than the table.

### Pass and To Close a Round:

A player may pass their turn if they have no card(s) to play or would prefer to pass. Once a player passes a turn, that player does not have another turn until the round ends.

When all players have played their respective turns, the round ends. The last player to play a hand begins the next round without any cards on the table.

### Eight-ender (8 Rule, 8 Giri):

A round ends when a player plays a hand containing an 8.

### 3 of Spades:

When the joker is played as a single card, a player may end the round by playing the 3 of spades.

### Revolution (Kakumei):

When a player plays a set (group [pair]) of four or more cards with the same number, or a sequence with five or more cards, a revolution occurs and the strengths of all cards are reversed until the end of the game.

### Lock (Tight, Shibari):

When a player follows the same suit as the play on the table, the round is tightened by the suit and all players must follow the same suit until the round ends.

### Special Titles (Rank, Mibun):

The first player with no cards left is the Daifugo, second is the Fugo, third is the Heimin, fourth is the Hinmin, and last is the Daihinmin. In the next game, the Daifugo hands two cards to the Daihinmin and the Fugo hands one card to the Hinmin. The choice of cards to hand to another player is arbitrary. However, the Daihinmin must yield two of their strongest cards to the Daifugo and the Hinmin gives their strongest card to the Fugo.