Playing Hanabi Near-Optimally

Bruno Bouzy

LIPADE, Université Paris Descartes, FRANCE,
bruno.bouzy@parisdescartes.fr

Abstract. This paper describes a study on the game of Hanabi, a multi-player cooperative card game in which a player sees the cards of the other players but not his own cards. Previous work using the hat principle reached near-optimal results for 5 players and 4 cards per player: the perfect score was reached 75% of times on average. In the current work, we develop HANNIBAL, a set of players, aiming at obtaining near-optimal results as well. Our best players use the hat principle and a depth-one search algorithm. For 5 players and 4 cards per player, the perfect score was reached 92% of times on average. In addition, by relaxing a debatable rule of Hanabi, we generalized the near-optimal results to other numbers of players and cards per player: the perfect score was reached 90% of times on average. Furthermore, for 2 players, the hat principle is useless, and we used a confidence player obtaining high quality results as well. Overall, this study shows that the game of Hanabi can be played near-optimally by the computer.

1 Introduction

Hanabi is a multi-player cooperative card game that received the 2013 best game award. All the players are in the same team. The goal is to reach a score as high as possible by building fireworks. A player can see the cards of the other players but he cannot see his own cards, which is the main particularity of the game. Hanabi had a great success among human players. Computer Hanabi also has a community and some work can be mentioned [10], [8], [5], [4]. [4] is the most significant. It is based on the hat principle [2] used in recreational mathematics. For the most common version of Hanabi with 5 players and 4 cards per player, Cox and his colleagues designed strategies that reach scores that are perfect 75% of times, by using the hat principle [4]. They used a restricted version of Hanabi in which a player is not allowed to inform a player about a color or a height of a card not belonging to his hand. This restriction is very debatable. This paper relaxes this restriction and uses the hat principle. Furthermore, it uses a tree search to improve the results of Cox. We developed HANNIBAL, a Hanabi playing program based of these features. The HANNIBAL's results generalize the previous results to other numbers of players and other numbers of cards per player. Moreover, with tree search, HANNIBAL's results enhance the previous results with, for example, perfect scores 92% of times for Hanabi with 5 players

and 4 cards per player. We claim that HANNIBAL plays Hanabi near-optimally. Since Hanabi is an imperfect information game, the results must be obtained by measuring average scores obtained on test sets that are as large as possible. Near-optimality means that the average scores obtained are not far from the optimal expected scores which are less than 25 and less than upper bounds estimated with average scores obtained by seer players.

The outline of the paper is the following. Section 2 defines the rules of Hanabi necessary to understand this paper. Section 3 gives the state of the art of computer Hanabi, and explains the essential idea of the hat principle. It is not possible to give all the details underlying the hat principle here without risking to misrepresent Cox's work. Therefore, the reader interested by these details can directly read Cox's paper. Section 4 is a debate concerning crucial rules according to which our work or Cox's work have very different outcomes. Section 5 lists the players we developed to perform the experiments. Before conclusion, section 6 gives the results of these experiments.

2 The game of Hanabi

The game of Hanabi is a multi-player and cooperative card game. The goal is to build "fireworks" with cards. There are five fireworks to build, each one with a specific color: red, blue, green, yellow or white. A firework has a height, an integer between 0 and 5, corresponding either to the height of the card situated on the top of the stack of the firework, or to 0 if the stack is empty. A card has a color (red, blue, green, yellow or white) and a height (1, 2, 3, 4 or 5). A card corresponds to the color and the height of a firework. There are 50 physical cards in total. For each color, there are ten physical cards: three 1, two 2, two 3, two 4, and one 5. Beforehand, the set of cards is shuffled and distributed to the players. The remaining cards are hidden in the deck.

There are several players. Let NP be the number of players. A player has a hand of cards. Let $NCPP$ be the number of cards per player. A player cannot see his own cards but he can see the cards of the other players.

There are several stacks: one stack for each firework, a deck of hidden cards and a stack of visible discarded cards. Moreover, there are eight blue tokens and three red tokens in a box. At the beginning, the height of the five fireworks is 0. The players move one after each other. There are three kinds of moves:

- playing a card,
- discarding a card,
- informing another player about one's hand.

To *play a card*, the player announces the card of his hand which he wants to play. If the card's height is one plus the height of the firework of the color of the card, then the card is added on top of the stack representing the firework, whose height is incremented by one. Otherwise, the card is discarded and the team of players receives a penalty: a red token is removed from the box. If the deck is not empty, the player takes the card on top of the deck to complete his hand.

To *discard a card*, the player announces the card he wants to discard. The card is put into the stack of discarded cards. This move is allowed if the number of blue tokens in the box is less than seven. In such a case, a blue token is moved into the box. If the deck is not empty, the player takes the card on top of the deck to complete his hand. The rule on the number of blue tokens forbidding to discard a card is debatable (see the discussion in section 4).

To *inform a player*, the informing player designates a player to inform with either a color value or a height value. If a color (respectively a height) is chosen, the informing player shows all the cards of the hand that have the corresponding color (respectively height). This move is allowed if the number of blue tokens in the box is positive. In such a case, a blue token is removed from the box. A rule forbidding to inform a player with a color or with a height not correponding to a card of the hand of the informed player can be used or not. For instance, this rule forbids to inform a player of his green cards when this player has no green card. This rule is very debatable (see the discussion in section 4).

The game continues while at least one red token remains in the box, and until each player has moved once after the deck has become empty. The score of a game is the sum of the heights of the fireworks. A game is perfect when the score reaches $5 \times 5 = 25$.

The interest of the game consists in balancing the moves adequately between giving information, discarding and playing. Playing a card increases the score by one point and uncovers one card from the deck: it can be considered as a good move. Discarding a card uncovers one card from the deck and adds one blue token into the box. Discarding an important card hinders reaching the maximal score. Informing a player gives him more knowledge on his cards but removes one blue token.

3 State of the art

The state of the art on Computer Hanabi is the following. This section describes previous work and the hat principle.

3.1 Previous work

Osawa [10] describes experiments with two players and five cards per player. Several strategies are described: the most sophisticated is the "self-recognition strategy", which includes an opponent model and produces an average score of 15.85. Van den Berghe [8] describes experiments with three players and five cards per player. Several strategies are described as well: the best one produces an average score of 15.4. Franz [5] describes experiments with four players and five cards per player performed with Monte-Carlo Tree Search [3], which yield an average score of 17. Cox and colleagues [4] describe very efficient strategies based on the hat principle [2], which yields an average score of 24.5 with five players and four cards (the standard version). However, this work has restrictions concerning the rules of the game which enable the method to work on the standard version only.

3.2 The hat principle

The hat principle [2] results in scores that reach 25 very often [4], which appears to be magic at a first glance. In this section, we use the recommendation strategy [4] to illustrate the hat principle in our own words. The idea underlying the hat principle is to represent the hand of a player with a "hat", i.e. a number h such that $0 <= h < H$. In the recommendation strategy, $H = NCPP \times 2$. The hat h of a player "recommends" a move to the player: when $h < NCPP$, the recommendation is "play card number h" starting from the left. Otherwise, the recommendation is "discard card number $h - NCPP$" starting from the left. There is a public recommendation program, named $RECOMPROG$, used by all players which outputs the hat of a given hand. A specific player sees the hands of the other players. Consequently, he can compute their hats with $RECOMPROG$. Communicating with the hat convention consists in using the information moves of Hanabi to transmit the sum of the hats that the player sees. When a player observes an information move performed by a given player, he can compute the value of his own hat by difference between the sum of hats transmitted within the information move and the sum of hats he sees (except the hat of the given player).

To make the hat convention work, there are technical details. Two public one-to-one mappings are used by all the players. With a code S such that $0 <= S < H$, $Code2Couple$ outputs a couple (B, I) where I is the information to send to player B by player A ("color Red" for instance). With a couple (B, I), $Couple2Code$ outputs a code S. When player A wants to give information, he computes S, the sum of the hats that he sees modulo H, and he informs with $Code2Couple(S) = (B, I)$. Therefore, the other players see (B, I) and they deduce $Couple2Code(B, I) = S$, the sum of the hats seen by A. Therefore, each player, different from A, seeing all the hats seen by A except his own hat, can compute the value of his own hat.

The hat principle is powerful in that an information move informs all the players at once, not only the targeted player. Therefore the blue tokens can be saved more frequently. The hat principle is well-known in recreational mathematics [2].

In the information strategy, the hat does not correspond to a recommended move but to possible values of the card with the highest playing probability [4]. Technically, each player uses the same public function that selects the unique card with the highest playing probability. The information strategy informs all the players at once about the possible values of their highest probability card. However, the information strategy needs room to be correctly described. For further details, we let the reader directly refer to the original paper. The information strategy is complex. It only applies when $NP - 1 >= NCPP$. This is the reason why Cox's results are limited to $NP = 5$ and $NCPP = 4$.

4 Rules

Relaxing one rule of Hanabi may lead to very different outcomes. The first rule to relax is allowing the players to see their own cards. Another rule to relax is the respect of the number of blue tokens: you may inform or discard whatever the number of blue tokens. Another rule to relax is allowing/forbidding to inform a player with a color or a height absent of his hand.

4.1 Seers and blue tokens

We call a *seer*, a player that can see his own cards but not the deck. The score obtained by a team of seers gives an upper bound on the score that could be obtained by the same team of players not seeing their own cards. Given that all hands are seen, a first design intuition is to remove the information moves and the blue tokens. However, since the seer player is designed for a fair comparison with normal players, it is actually fair and relevant to keep the respect of blue tokens for seer players as well. In such case, an information move does not add actual information but decrements the number of blue tokens allowing a discard move at the next turn.

4.2 Informing with any color or any height, or not ?

Cox's work assumes that you cannot inform a player with a color information or rank information not part of his hand [4]. For instance, if a player has no green card, you cannot inform him with "color green: empty set". This assumption is a strong one. Let CH be the kind of information of an information message, *color* or *height*: it has two values only. Given $NP-1$ players are able to receive the information, there are $2 \times (NP-1)$ values of code which can be sent by an information move. For instance, with $NP = 5$, the code may have 8 values, which is adapted to the recommendation strategy when $NCPP = 4$. However, with 8 values, you cannot code the 25 values of a card, and the information strategy cannot be simple in this context [4].

If the rule of the game permits to inform a player with any color and any rank (i.e. a color or a rank possibly absent of a hand), this gives 10 values contained in a message sent to a given player (5 heights plus 5 colors yield 10 possibilities). When considering the $NP-1$ receivers of the message, this gives $10 \times (NP-1)$ values of code. With $NP > 3$, the number of values of code is superior to 25, the number of card values. Therefore, with $NP > 3$, the hat of a hand can be defined to be the *exact* value of a specific card, which simplifies the information strategy. In Cox's work, the exact value of a card cannot be transmitted at once, and a complicated machinery solves this issue. In our work, we avoid this complication by assuming that informing with any color or any height is permitted.

Of course, there is a debate for or against this rule. First, the game set does not mention whether this rule must be on or off, which may open the debate. Secondly, Wikipedia [11], explicitly says that any color and any rank are *allowed*. Thirdly, a translation [9] of the German rules of Hanabi on Abascusspiele [1] also

explicitly says that any color and any rank are *allowed*. Fourthly, [4] says that any color and any rank are *forbidden*. In this paper, we assume a player is *allowed* to inform with any color and any rank.

5 Players

This section presents the players developed in our program HANNIBAL. There are knowledge based simulators that play a game instantly:

- a certainty player,
- a confidence player,
- a hat recommendation player,
- a hat information player,
- a seer player.

Furthermore, there is a player that can be launched by using a simulator of the previous list:

- a tree search player.

5.1 The certainty player

The certainty player uses the following convention. While information has to be given on playable cards and useless cards, give the information. Play a card as soon as this card is playable *with certainty*. Discard a card as soon as this card is discardable *with certainty*. When blue tokens are missing, discard the oldest card of your hand. The strategy resulting from these principles is slow in that a card needs to be informed twice - color and height - before being played or discarded.

5.2 The confidence player

To speed up the previous strategy, the idea of the confidence convention is, as far as possible, to inform cards once before being played or discarded. When a player explicitly informs another player about cards, he also sends an implicit information to the informed player meaning that the targeted cards can be either played or discarded on the current board. The informed player must discard the card if he can conclude by himself that the card has to be discarded. Otherwise, the informed player can *play the card with confidence*. When blue tokens are missing, discard the oldest card of your hand. Compared to the certainty convention, this convention accelerates the playing process, the discarding process, and the blue tokens are spent less often.

5.3 The hat recommendation player

For a detailed description of the whole recommendation strategy see [4]. We did our best so that our recommendation strategy mentioned in section 3.2 be identical to Cox's recommendation strategy.

5.4 The hat information player

See the information strategy of [4]. Like in [4], the first key concept is the playing probability of a card. The playing probability of a card is computed given the public information on this card. Since this computation uses public information only, it can be performed by any player. The card with the greatest playing probability in the hand of a player is the card targeted by the information strategy for this player. The second key concept is the hat idea described in section 3.2. Our hat information player is a simplication of Cox's information strategy because the rule forbidding informing about absent cards is off. Consequently, in our work, the hat of a player corresponds to the value of the targeted card of the player.

5.5 The seer player

The seer player sees his own cards but not the cards of the deck. In our work, we designed two seer strategies: the recommendation program, $RECOMPROG$, of the recommendation strategy mentioned in section 3.2 enhanced with the blue token respect and information moves, and the information strategy of section 3.2 assuming that the cards are seen.

5.6 The tree search player

The tree search player mainly follows the expectimax algorithm [7]. First, let us describe the main similarities. It is a tree search at a fixed depth. One depth includes a layer of max nodes and a layer of chance nodes. A max node corresponds to a state in which a player has to move. A chance node corresponds to an action-state in which a card in the hand of the player has to be revealed (in the cases of playing and discarding moves only) and the card on top of the deck has to be revealed. The tree search player must be launched with a given depth $DEPTH$, and with a number of card distributions NCD following a chance node. NCD is also the number of nodes following a chance node. In practice, $DEPTH$ equals one or two, and NCD equals 10^x with $1 \leq x \leq 4$.

Our tree search player has two main differences with the expectimax algorithm. First, instead of using a probability distribution of next possible futures, our tree search uses NCD actual futures, each of them corresponding to one actual card distribution. In a given action-state (or chance node), given the visible cards and the past actions, the tree search player needs a card distribution for hidden cards. A card distribution is a solution of an assignment problem [6]. This solution can be found in polynomial time by the Hungarian method [6]. Therefore, so as to generate a random distribution of cards that respects the visible cards and the past actions, our tree search player uses the Hungarian method. Secondly, at a leaf node, the value of the node is the outcome of a knowledge-based simulation, and not the result of an evaluation function call.

Table 1: For $NP = 2, 3, 4, 5$ (one line for each value). For each line and from left to right: mean values obtained by the certainty player, the confidence player, the hat recommendation player and the hat information player for $NCPP = 3, 4, 5$.

	Certainty			Confidence			Hat Recommend.			Hat information		
	3	4	5	3	4	5	3	4	5	3	4	5
2	10.31	10.71	11.10	**16.89**	16.69	15.85	15.78	**16.92**	**17.80**	5.95	6.42	6.72
3	12.86	13.04	13.54	19.37	19.18	17.87	**22.82**	**23.78**	**23.84**	18.74	19.45	18.88
4	14.38	14.74	14.09	20.28	19.66	17.91	23.25	23.48	22.79	**24.27**	**24.66**	**24.40**
5	15.21	14.36	12.77	20.57	19.24	16.81	23.24	22.61	20.99	**24.57**	**24.74**	**24.30**

6 Experiments

In this section, we describe the experiments performed by HANNIBAL on the game of Hanabi with a homogeneous team of players. Since the team is homogeneous, the term player refers either to an individual player belonging to a team or to a whole team. An experiment is a set of NG games with NP players and $NCPP$ cards per player with $2 \leq NP \leq 5$ and $3 \leq NCPP \leq 5$. Each game starts on a card distribution that corresponds to a specific seed $Seed$ with $1 \leq Seed \leq NG$. A game ends with a score $Score$ with $0 \leq Score \leq 25$. An experiment result is the mean value of the scores obtained on the NG games, and a standard deviation. The minimal and maximal scores can be output as well. In some specific conditions where the players are near-optimal, the histogram of the scores can be built, and the percentage of 25 can be relevant information as well. For the tree search player, $NG = 100$. Otherwise, $NG = 10,000$. We used a 3 Ghz computer.

6.1 The knowledge-based players

In this section, we provide the results obtained by the knowledge-based players, i.e. the certainty player, the confidence player, the hat recommendation player and the hat information player.

The first three columns of Table 1 show the mean values obtained by the certainty player. $NG = 10,000$. The scores obtained are superior to 10 on average. This is a first result far from the maximal score of 25.

The next three columns of Table 1 show the mean values obtained by the confidence player. $NG = 10,000$. The scores obtained are superior to 15 on average. For some values of NP and $NCPP$, the scores reach 20 on average. This is a second result that shortens the distance to the maximal score of 25. This result underlines the domination of the confidence principle over the certainty principle.

The next three columns of Table 1 show the mean values obtained by the hat recommendation player. $NG = 10,000$. For $NP = 2$, the scores are greater than 15 on average and remain in the same range as the scores obtained by

Table 2: Histogram of scores obtained for $NP = 5$ and $NCPP = 4$.

Score	19	20	21	22	23	24	25
%	0.01	0.05	0.17	1.19	3.62	13.66	81.30

the confidence player. This fact is explained by the relative uselessness of the hat principle for 2 players. For $NP \geq 3$, the scores obtained range around 22 or 23, which represents a large improvement. The scores are not far from 25. This fact is explained by the usefulness of the hat idea for many players. A hat information informs many players in one move. The information moves can be used less often. It is worth noting that [4] obtains 23.0 on average for $NP = 5$ and $NCPP = 4$, where our player obtains 22.61 only. The small difference between the two results can be explained by a possible implementation difference that we could not reduce and/or by a difference of test set.

The last three columns of Table 1 show the mean values obtained by the hat information player. $NG = 10,000$. For $NP = 2$, the scores remain around 6, which is very bad actually. For $NP = 3$, the scores remain around 19, which is comparable to the scores of the confidence player. Our adaption of the hat information player is designed for $NP \geq 4$ only. The scores are greater than 24 on average, which represents another large improvement. The average scores are very near from 25. To this extent, showing the histogram of actual scores becomes relevant. Table 2 shows the histogram of the actual scores obtained for $NP = 5$ and $NCPP = 4$. Our hat information player is near-optimal in that he reaches 25 more than 81% of the times. This result is better than the result of [4] (75%). This can be explained by the fact we have relaxed the constraint forbidding to inform about a rank or a color which is not in the hand of the player to be informed (see the discussion in section 4). Here, we reached a point where the hat principle is highlighted by near-optimal results. The next question is to see how near from optimality these results are. An experiment with seer players in the next section will give the beginning of an answer.

6.2 The seer players

Our first seer player is $RECOMPROG$ (see section 3.2). Our second seer player is the decision program of the information strategy (see section 3.2). The three columns on the left of Table 3 show the mean values obtained by our first seer player. $NG = 10,000$. The results are excellent.

The three columns on the right of Table 3 show the mean values obtained by our second seer player. $NG = 10,000$. The results are excellent as well. For $NCPP = 3$, they are slightly better than those obtained by the first seer player. For $NCPP = 4$ or $NCPP = 5$, they are almost equal to those obtained by the first seer player. The informative point of these tables is to show results which can be hardly surpassed by normal players. Their contents have to be compared

Table 3: Mean values obtained by the seer players being *RECOMPROG* (left) or the decision program of the hat information strategy (right).

	Seer players					
	RECOMPROG			hat info decision		
	3	4	5	3	4	5
2	22.10	24.05	24.73	**23.54**	**24.47**	**24.76**
3	24.14	**24.86**	**24.95**	**24.49**	24.80	24.89
4	24.58	**24.91**	**24.94**	**24.69**	24.86	24.90
5	24.69	**24.86**	**24.85**	**24.75**	24.83	24.60

Table 4: Mean values obtained by tree search players at depth one using the confidence player, the hat recommendation player, the hat information player, or *RECOMPROG* (a seer) as evaluator. $NG = 100$.

	Tree search players											
	Confidence			Hat Recommend.			Hat information			*RECOMPROG*		
	3	4	5	3	4	5	3	4	5	3	4	5
2	**19.22**	**19.42**	**18.98**	16.40	17.38	18.53				23.10	24.46	24.91
3	20.73	21.08	20.44	**23.96**	**24.56**	**24.70**				24.62	24.97	25.00
4	21.55	21.05	19.67	24.34	24.60	24.45	**24.72**	**24.96**	**24.91**	24.91	25.00	24.99
5	22.01	20.41	17.97	24.26	24.30	22.68	**24.85**	**24.92**	**24.76**	24.96	24.98	24.96

with the contents of Table 1. This comparison shows that the normal hat players are not far from their maximal scores.

6.3 The tree search player

Table 4 shows the mean values obtained by the tree search player using - from left to right - the confidence player, the hat recommendation player, the hat information player or the seer player *RECOMPROG* as evaluator. $NG = 100$. $DEPTH = 1$. $NCD = 10,000$ i.e. $x = 4$. We used a 3 Ghz computer and let 10 minutes of thinking time, which corresponds to 10 seconds per move on average.

On the right, the table shows that the tree search player using the seer player (being *RECOMPROG*) produces near-optimal results. Over the $NG = 100$ games, a 25.00 in a cell means that the player succeeds a 25 for all games, and a 24.99 means that the actual scores are always 25 except for one of them which is 24. This specific player is a cheater but gives a measure of the hardness of a card distribution. Those results also indicate that our card distributions are never with many 1 of a given color at the bottom of the deck. We have tried to use the decision program of the hat information player as a seer used by the tree search player, but, surprisingly, the results were not as good as those in the table whatever the values of NP and $NCPP$.

For players not seeing their own cards - the real game - the results are excellent. For $NP \geq 4$, the best results are obtained by the tree search player using the hat information player. For $NP = 5$ and $NCPP = 4$, the average score is 24.92 meaning that, over the 100 games, 92 of them end up with a 25 and 8 of them with a 24. The perfect scores are obtained 72%, 96%, 91%, 85%, 92%, or 76% of the times on the test set. These best results obtained by the normal players have to be compared with the results obtained by the tree search player using $RECOMPROG$ a seer player. For $NP = 4$ or $NP = 5$, the perfect scores of our tree search seers are obtained 91%, 100%, 99%, 96%, 98%, or 96% of the times on the test set. This comparison shows that the normal hat players are not far from their maximal scores.

This result is better than the result of [4]. However, conversely to a hat information player, a tree search player uses a significant amount of CPU time. The longer the CPU time the better the results. The results given here are obtained with one game lasting about 10 minutes and one move decision lasting 10 seconds. The tree search player develops a tree at depth one. We have tried $DEPTH = 2$ with $NCD = 100$ but the results were not better. Actually, the variance on the simulation outcomes is high due to the hidden card drawn from the deck. A depth-one search with $NCD = 10,000$ is more accurate than a depth-two search with $NCD = 100$. Furthermore, for the same cause, under our time constraints, we believe that MCTS which is designed to develop deep trees would be less accurate than our depth-one search.

For $NP = 3$, the best results are obtained by the tree search player using the hat recommendation player. For $NP = 2$, the best results are obtained by the tree search player using the confidence player.

$NP = 2$ or $NP = 3$ has no meaning for our hat information strategy because this strategy needs $10 \times (NP - 1) >= 25$ to work. This explains the empty cells in Table 4.

7 Conclusion

In this paper, we described a work on the game of Hanabi. We developed HANNIBAL, a set of players, each player being either a knowledge-based simulator or a tree search player using a simulator. The simulators use different kinds of knowledge: certainty, confidence or hat principle. We improved the results obtained by [4] for $NP = 5$ and $NCPP = 4$ with 92% on average of perfect scores (instead of 75%). This was done by using the hat recommendation $RECOMPROG$ of [4] used by a depth-one tree search player with 10 minutes of thinking time on a 3 Ghz computer. Moreover, we generalized the results for $NP \geq 3$ whatever $NCPP$ with near-optimal results (90% of perfect scores). These results are obtained with a depth-one tree search using the hat recommendation player as simulator. For $NP = 2$, we obtained results with a depth-one tree search using a confidence player as simulator. These results assume that a player is allowed to inform another player with any color or any height whatever the cards of the informed player. As far as we know, all these results surpass the previous ones,

when they exist. We also developed seer players that obtained near-optimal results giving upper bounds to the results of normal players. Our results show that Hanabi is not a difficult game for the computer, which can deal with the hat principle easily.

In the current work, we used depth-one tree search associated with playing simulators, and the resulting move costs computing time. Building a state value function with temporal difference learning, or an action value function with Q learning, both based on a neural network as approximator is an interesting direction to investigate. With such action value function, the player could play his move instantly and could reach a playing level comparable to the level reached in the current work. A state value function could be used in a tree search as well, possibly improving the current results. However, beyond the fact of improving the playing level of the current work, investigating the neural network approach is also an opportunity when considering the convention used by the Hanabi players (certainty, confidence, hat convention, or any other convention). A specific convention could be learnt by the network or better: uncovered by the network, which is very exciting and challenging.

Since the particularity of Hanabi is cooperation and hidden information, working on other card games with competition and hidden information, such as Hearts, Poker and Bridge, is another motivating direction to investigate.

References

1. Abacusspiele. Abacusspiele, 2017. http://www.abacusspiele.de/en/spiele/hanabi/.
2. E. Brown and J. Tanton. A dozen of hat problems. *Math Horizons*, 16(4):22–25, 2009.
3. Cameron Browne, Edward Powley, Daniel Whitehouse, Simon Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte-Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1), 2012.
4. Christopher Cox, Jessica De Silva, Philip Deorsay, Franklin H.J. Kenter, Troy Retter, and Josh Tobin. How to make the perfect fireworks display: Two strategies for hanabi. *Mathematics Magazine*, 88(5):323–336, December 2015.
5. Robin Franz. Modeling metareasoning in games. Master's thesis, CogMaster, Paris Dauphine University, 2016.
6. Kuhn. The hungarian method for the assignment problem. *Naval research logistics Quarterly*, 2(1-2):83–97, 1955.
7. D. Michie. Game-playing and game-learning automata. *Advances in Programming and Non-Numerical Computation*, pages 183–200, 1966. L. Fox (ed.).
8. W.A. Kosters M.J.H. van den Bergh and F. Spieksma. Aspects of the cooperative card game hanabi. In *Proceedings BNAIC 2016*, pages 25–32, 2016.
9. Wade Nelson. Hanabi german rules translated from abacusspiele, 2017. https://boardgamegeek.com/thread/886616/english-rules-translation.
10. Hirotaka Osawa. Solving Hanabi: Estimating hands by opponents actions in cooperative game with incomplete information. In *Workshop at AAAI 2015: Computer Poker and Imperfect Information*, pages 37–43, 2015.
11. Wikipedia. Hanabi card game, 2017. https://en.wikipedia.org/wiki/Hanabi.