Aspects of the Cooperative Card Game Hanabi

Mark J.H. van den Bergh [a b] Walter A. Kosters [a] Flora M. Spieksma [b]

[a] *Leiden Institute of Advanced Computer Science, Universiteit Leiden*
[b] *Mathematical Institute, Universiteit Leiden*

Abstract

We examine the cooperative card game HANABI. Players can only see the cards of the other players, but not their own. Using hints partial information can be revealed. We show some combinatorial properties, and develop AI (Artificial Intelligence) players that use rule-based and Monte Carlo methods.

1 Introduction

The game of HANABI, meaning "fire flower" or "fireworks" in Japanese, is a cooperative card game which requires the players to combine efforts in order to achieve the highest possible score. Designed by Antoine Bauza in 2011 and published by R & R Games [7] (see Figure 1), among others, it is designated as a game in the categories "cooperative play" and "hand management" by the game analysis site BOARDGAMEGEEK [3]. The goal of the game is simple: play out several sequences of cards in the right order. The catch, however, is that the players can only see the cards in other player's hands and not their own; information has to be gathered by a system of hints that reveal partial information.

Figure 1: HANABI, as sold by R & R Games [7].

A game in the named categories which is somewhat similar in gameplay is THE GAME designed by Steffen Benndorf [3]. Though leading to interesting combinatorial and strategic questions, there is not much to be found in the literature on this kind of games. More is known about another game which bears similarity to Hanabi in view of its dealing with hints: BRIDGE; some issues here touch upon research as in [5]. For instance, even *not* giving a hint might reveal some information. Moreover, it is possible to use artificial hints having non-natural meaning, like conventions in BRIDGE. Yet another game which gives rise to theoretical problems comparable to those inspired by Hanabi, because of its similarity in nature, is SOLITAIRE; as an example, NP-completeness results for HANABI can be found in [1].

This paper examines some interesting properties of the game, but is far from being complete. In fact, we show complicated mathematical behavior for a one-player version without hints, and provide an exploratory examination of some artificial players (cf. [6, 2]), including rule-based and Monte Carlo versions.

We start with a comprehensive explanation of the rules of HANABI in Section 2. We then deal with two main questions. In Section 3, we consider the playability of a game of HANABI: given a certain start configuration, is it possible to obtain a maximal score when playing perfectly? We will address some theoretical issues. Next, in Section 4, we look for a good strategy for any arbitrary game. Among others, we consider Monte Carlo methods which seem promising. In Section 5, we conclude with a summary of the given results as well as some interesting open questions.

2 Game Rules

The classic game of HANABI is played with a stack of $N = 50$ cards. Every card has one out of $C = 5$ colors — blue (B), red (R), green (G), yellow (Y) or white (W) — and a value between 1 and $k = 5$. In the classic game, for every color there are three 1s, two 2s, 3s and 4s and one 5, hence fifty cards in total. At the start of the game, the stack is shuffled and a hand of cards is dealt to each player: $R = 5$ cards are given to every player if $P = 2$ or $P = 3$ persons are playing, and $R = 4$ cards are dealt in games with $P = 4$ or $P = 5$ players. Now, every player picks up their hand in such a way that the other players can see them, but they themselves cannot. The rest of the cards forms the face-down stack.

The goal of the game is to create C stacks of cards going from 1 through k on the table, one of each color. To do so, players take turns, choosing exactly one of the three following actions every turn:

- give a hint,
- discard a card,
- play a card.

At the start of the game, a pool of $H = 8$ hint tokens is available to the players. To give a hint to another player, one token must be removed from this pool. If there are no tokens left, this action cannot be chosen. Giving a hint is done by either pointing out all cards (perhaps zero) of a certain color or all cards of a certain value in the hand of one other player. This is explained most easily by considering an example hand like $(\mathrm{R3}, \mathrm{W1}, \mathrm{B1}, \mathrm{R4}, \mathrm{B5})$. A hint may be expended to point out the position of the $\mathrm{W1}$ and $\mathrm{B1}$, telling the player that *these two cards are 1s*. One might also point out the $\mathrm{R3}$ and $\mathrm{R4}$ by telling that *these cards are red*. Also, pointing out $\mathrm{B5}$ by saying *this is a 5* is fine. However, one cannot point out $\mathrm{B5}$ by telling that *this card is blue*, because $\mathrm{B1}$ must then also be pointed at. One is allowed to tell the player that *there are no green cards in the hand*, as this hint effectively points out all (zero) green cards.

The second possible action is to discard a card: the active player takes a card from his/her hand (without first looking at it) and announces that it will be discarded. It is then put face-up in the discard pile, which may be viewed by all players at any point in the game. Once in the discard pile, a card will never re-enter the game. A new card is now taken from the face-down stack to replenish the hand and as an added bonus, a hint token is added to the available hints pool. Therefore, if there are already H hints available, this action cannot be chosen.

Finally, one may choose to try to play a card much in the same way as discarding a card. However, the players now look whether the card can be added to one of the stacks on the table. This is again best clarified by an example. Suppose the following stacks are on the table:

R1	G1	W1
R2		W2
		W3

It is now possible to play a $\mathrm{R3}$, which can be appended to the leftmost stack. Similarly, a $\mathrm{G2}$ or $\mathrm{W4}$ would be fine. It is also possible to play a $\mathrm{Y1}$ or $\mathrm{B1}$, starting a new stack. However, one may for example not play a $\mathrm{R4}$ (as a $\mathrm{R3}$ is first needed) or $\mathrm{R2}$ (as this is already on the table). Moreover, another $\mathrm{R1}$, $\mathrm{G1}$ or $\mathrm{W1}$ cannot be played as there may only be one stack of every color on the table at any time. If a card is successfully played, it is put in the fitting position on the table. If a card turns out not to be playable, it is moved to the discard pile and the players score one error. In either case, the active player draws a new card from the stack, but no hint token is added.

In order to show that there is more to a hint than just its plain meaning, consider the following example. If a player has only one 2, the G2, a hint pointing at this card might in this situation be considered containing the message "come on, play this card".

The game ends when one of three conditions is met. First, if $E = 3$ errors are made by trying to play a card, the game ends with the lowest possible score of 0. Second, if the C-th stack is completed by playing a card, i.e., if there is a sequence of 1 through k of every color on the table, the game ends with the highest possible score of $C \cdot k$ (for the classic game: 25). Finally, when the last card of the face-down stack is drawn, every player gets one more turn, including the player that drew the last card. The achieved score is then determined by adding the highest number of every stack on the table. In the previous example, a score of $2 + 1 + 3 = 6$ points would be obtained. Apart from just aiming at the highest score, one might also focus on the 0–1 target of obtaining the maximal score.

Naturally, the rules for classic HANABI can easily be generalized and extended. One can for example alter the parameters mentioned above, see Section 3. A more formal approach can be found in [2].

3 Playability

In this section, we will address the first main question: given an initial configuration of a game of HANABI, is it possible to obtain the maximum score if playing perfectly? An initial configuration of a game of HANABI (or simply a game of HANABI) is called *playable* if the maximum score can be achieved. We first turn to a theoretical approach involving combinatorics; a perhaps more practical approach using dynamic programming is presented in [1] and also in [2].

As in [1], we consider the simplified situation where the players can also see their own cards (making the hints system and errors obsolete). We give a result for the one-player version, with $R = 1$, called SINGLEHANABI: a player must immediately play or discard the newly received card. We consider only one color, but the number of cards of each value can be arbitrary.

Fix an integer $k \geq 1$. Let $x = (x_1, x_2, \ldots, x_k)$ be a vector with k non-negative integers $x_1, x_2, \ldots, x_k$, and let $F(x) = F(x_1, x_2, \ldots, x_k)$ denote the number of ordered sequences of length $N = N(x) = x_1 + x_2 + \ldots + x_k$ with x_1 occurrences of the integer 1, x_2 occurrences of the integer 2, …, x_k occurrences of the integer k, *without* a subsequence 1–2–…–k: the so-called *bad* sequences; the others are called *good*. Note that a subsequence is not necessarily consecutive; e.g., the sequence 1–3–2–3 has 1–3–2 and 1–2–3 as subsequence, but not 2–3–1. So $F(x_1, x_2, \ldots, x_k)$ is the number of unplayable SINGLEHANABI games, having x_i cards of value i ($1 \leq i \leq k$): there is no subsequence 1–2–…–k that would allow the player, who has "no memory", to immediately play these k "cards" in order.

If for some i with $1 \leq i \leq k$ we have $x_i = 0$, then we know that $F(x) = \binom{N}{x_1, x_2, \ldots, x_k, N - \sum_{j=1}^{k} x_j}$, since all sequences are bad in that case.

We also note that F is symmetric in its arguments. Indeed, we construct a bijection between good sequences with interchanged numbers of, e.g., 1s and 2s: $x_1 \leftrightarrow x_2$. To do this, view every good sequence in parts: the part up to but not including the first 1, the part between this 1 and the next 2, etc. Now, the required bijection is given by swapping the first and second part and changing all 1s in the parts to 2s and vice versa. We clarify this by an example, in which the numbers which divide the parts are shown in bold:

$$24\mathbf{1}313413132\mathbf{2}1\mathbf{3}32\mathbf{4}21421 \leftrightarrow 32342323\mathbf{1}14\mathbf{2}2\mathbf{3}31\mathbf{4}12412$$

Note that the resulting sequence contains the proper amount of every number. Moreover, applying the proposed bijection twice results in the original sequence, which shows that it is indeed bijective.

Inspired by a discussion by anonymous contributors at STACKEXCHANGE [9], our main result in this section is:

Theorem We have

$$F(x) = \sum_{\substack{y \prec x \\ |y| \leq k-2}} a(N, y)$$

Here we denote $y \prec x$ if the ordered sequence y with k non-negative integers satisfies $y_i \leq x_i$ for all i with $1 \leq i \leq k$. Furthermore, $|y|$ denotes the number of non-zero elements in y. We put $a(n, y) = (-1)^{|y|} \binom{n}{y}^* (k - |y| - 1)^{n - s(y)}$. The multinomial coefficient $\binom{n}{y}^*$ is defined as follows: $\binom{n}{y}^* = \binom{n}{y_1 \ominus 1, y_2 \ominus 1, \ldots, y_k \ominus 1, n - s(y)} = n!/((y_1 \ominus 1)!(y_2 \ominus 1)! \ldots (y_k \ominus 1)!(n - s(y))!)$ (the bottom last term in the multinomial coefficient, here $n - s(y)$, is often omitted by convention) with $s(y) = \sum_{i=1}^{k}(y_i \ominus 1)$, where we used $t \ominus 1 = \max(t-1, 0)$.

The equation can also be written as

$$F(x) = \sum_{\ell=0}^{k-2} (-1)^{\ell} \sum_{\substack{y \prec x \\ |y|=\ell}} \binom{N}{y}^{*} (k-\ell-1)^{N-s(y)}$$

For example, with $k = 3$ and $\ell = |y| = 1$, in the computation of $F(2,3,1)$ we encounter sequences $(1,0,0)$, $(2,0,0)$, $(0,1,0)$, $(0,2,0)$, $(0,3,0)$ and $(0,1,0)$; and, e.g., $\binom{6}{(0,2,0)}^{*} = \binom{6}{0,1,0,5} = \binom{6}{1,5} = \binom{6}{1} = 6$. The term with $\ell = |y| = 0$ equals $(k-1)^N$. Note that $F(x_1, x_2)$ evaluates to 1, as expected. By the way, $F(x_1)$ is 0, if $x_1 > 0$.

Proof The theorem can be proven through the obvious recurrence (for $x_1 > 0$)

$$\begin{aligned} F(x_1, x_2, \ldots, x_k) \quad &= \quad \binom{N-1}{x_1-1} F(x_2, \ldots, x_k) + \\ &\quad + F(x_1, x_2-1, \ldots, x_k) + \ldots + F(x_1, x_2, \ldots, x_k - 1) \end{aligned}$$

where we interpret a term as 0 if one of its arguments is negative. The respective terms count bad sequences that start with a 1, with a 2, …, with a k.

We first note that $F(1, \overbrace{0, \ldots, 0}^{k-1 \text{ times}})$ equals 0 if $k = 1$, and equals 1 if $k > 1$. And $F(0, 0, \ldots, 0) = 1$. This is the basis for an inductive proof, with respect to $\langle k, N \rangle$.

Using the symmetry of F in its arguments, we may assume that $x_1 > 0$. If $x_2 > 0$ we compute (analogous for the other terms):

$$F(x_1, x_2-1, \ldots, x_k) = \sum_{\substack{y \prec x;\, y_2 < x_2 \\ |y| \le k-2}} a(N-1, y) = \sum_{\substack{y \prec x;\, y_2 = 0 \\ |y| \le k-2}} a(N-1, y) \; + \sum_{\substack{y \prec x;\, y_2 \neq 0 \\ |y| \le k-2}} a(N, y) \frac{y_2 - 1}{N}$$

Now we look at a fixed y, and combine all contributions of the $k-1$ terms. If $y_1 = 0$ we arrive at $a(N-1, y)(k - |y| - 1) + a(N, y)s(y)/N = a(N, y)$. However, if y has non-zero y_1, we have to be more careful. We then still arrive at $a(N, y)$, but now with an additional term $a(N-1, y) - a(N, y)(y_1 - 1)/N$. If we let y_1 increase from 1 to x_1 (where we keep the other elements from y unchanged), these terms telescope to $a(N-1, y')$ with $y'_1 = x_1$ and $y'_i = y_i$ $(1 < i \le k)$. We rewrite

$$\binom{N-1}{x_1-1} F(x_2, \ldots, x_k) = -\sum_{\substack{y \prec x;\, y_1 = x_1 \\ |y| \le k-2}} a(N-1, y)$$

which exactly cancels the remaining terms.

If we happen to have $x_2 = 0$, the argument above remains valid. Indeed,

$$\sum_{\substack{y \prec x \\ |y| \le k-2}} a(N-1, y) = \sum_{\substack{y \prec x;\, y_1 < x_1 \\ |y| \le k-2}} a(N-1, y) + \sum_{\substack{y \prec x;\, y_1 = x_1 \\ |y| \le k-2}} a(N-1, y)$$

is equal to 0, since the first term from the right hand side in that case equals

$$F(x_1 - 1, 0, x_3, \ldots, x_k) = \binom{N-1}{x_1 - 1, x_3, \ldots, x_k}$$

whereas the second equals

$$-\binom{N-1}{x_1-1} F(0, x_3, \ldots, x_k) = -\binom{N-1}{x_1-1}\binom{N-x_1}{x_3, \ldots, x_k} = -\binom{N-1}{x_1-1, x_3, \ldots, x_k}$$

thereby completing the proof. □

One consequence is that

$$F(\overbrace{1, 1, \ldots, 1}^{k \text{ times}}) = k! - 1 = \sum_{\ell=0}^{k-2} (-1)^{\ell} \binom{k}{\ell} (k - \ell - 1)^k$$

which happens to be a special case of a formula from [8]. In this same category, another special case is

$$F(\overbrace{4, 4, \ldots, 4}^{13 \text{ times}}) = \sum_{\ell=0}^{11} (-1)^{\ell} \binom{13}{\ell} \sum_{i_1=0}^{3} \cdots \sum_{i_\ell=0}^{3} \binom{52}{i_1, \ldots, i_\ell} (12 - \ell)^{52 - i_1 - \ldots - i_\ell}$$

which evaluates to 91973270026324484565579418350194489655582912237019 (a prime number approximately equal to $9 \cdot 10^{49}$) using a straightforward Python program, that takes a few seconds. Therefore, the probability that a shuffled deck of standard playing cards contains a full increasing subsequence (from ace to king, disregarding suits) turns out to be equal to $1 - F(4, 4, \ldots, 4)/(52!/4!^{13}) \approx 0.000554$, which is a folklore result.

4 Strategies

We will now consider the second main question: what is a good strategy? In this section, a *strategy* is a means to determine which action to take in any given state of the game. We will look at games with $P = 3$ players in which every player has a hand size of $R = 5$ as in the classic game rules in Section 2. We call a card *useful* if it can be appended to one of the stacks on the table at the current moment, and otherwise *useless* — the latter term perhaps being somewhat preliminary.

In [6], strategies are considered in which the players try to estimate their hand by analyzing actions of other players. This analysis is for the two-player version, and relies heavily on online learning. The author claims an average result of 15.85 points. Here, we will try two other types of strategies, and explore their potential. The first one implements several rules of thumb that tend to come up quickly in human play: a rule-based strategy. The second is an implementation of a basic Monte Carlo strategy. Note that it is hard to determine the maximal score that can be achieved for any given game, see also [1].

For the first strategy, every player acts according to the following preset rules:

1. If there is a card in my hand of which I am “certain enough” that it can be played, I play it.
2. Otherwise, if there is a card in my hand of which I am “certain enough” that it is useless, I discard it.
3. Otherwise, if there is a hint token available, I give a hint.
4. Otherwise, I discard a card.

In this framework, there are several parameters to be determined. First, we may choose the definition of “certain enough” in steps 1 and 2: we let $\omega_{\rm p}, \omega_{\rm d} \in [0, 1]$ be the thresholds above which a player knowing that the probability of a card being useful, resp. useless, exceeds the threshold, it is played, resp. discarded, in step 1, resp. 2. Moreover, one can choose whether or not to take any risk to end the game on three errors by prohibiting to play cards which are not certainly useful after having made two errors; this is referred to as “safe play”.

In step 3, we can follow different guidelines which determine the hint to be given. We consider four of them: #1) random; #2) giving a hint that gives information on the largest number of cards; #3) giving a hint on the next useful card in sight or on the largest number of cards if no useful card is seen; or #4) giving a hint on the next useful card or on the next useless card if no useful card is available or otherwise on the largest number of cards.

Finally, in step 4, we choose from four different rules by which the card to be discarded is chosen. There are: #1) random; #2) discarding the card of which it is most certain that it is useless; #3) discarding the card which has been stored in hand the longest; or #4) discarding the card of which it is most certain that it is not absolutely necessary to complete all stacks.

In addition to these choices, we also explore whether it is profitable to sometimes swap the order of steps 3 and 4. We let $\omega_{\rm h} \in [0, 1]$ be the probability that this is done during a turn.

To test the various settings of the parameters, we let three players using the same strategy play 10,000 different starting configurations. Every configuration is played ten times in order to account for the randomness in the strategies. A test run of this kind takes approximately one minute on a computer with an Intel i7 2.3GHz core and 6 GB of RAM.

First, we take $\omega_{\mathrm{d}} = \omega_{\mathrm{h}} = 1.0$, and vary ω_{p} in $\{0.5, 0.6, 0.7\}$. The average scores for all sixteen combinations of hint and discard rules are shown in Table 1. It is apparent that the third hint rule, giving a hint on the next useful card or otherwise on the most cards, is dominant for this setting of the parameters. Surprisingly, discard rules #1 and #2 are about as effective, meaning that discarding randomly is competitive with discarding the card of which we most think it is useless.

		ω_{p}	Hint rule			
			#1	#2	#3	#4
Discard rule	#1	0.7	6.3	13.5	14.5	11.9
		0.6	7.1	13.9	15.3	12.7
		0.5	7.3	13.8	15.2	12.5
	#2	0.7	6.6	13.8	14.5	12.0
		0.6	7.4	14.4	**15.4**	12.7
		0.5	7.5	14.2	15.3	12.6
	#3	0.7	5.8	13.1	14.1	11.5
		0.6	6.6	13.5	14.9	12.3
		0.5	6.9	13.4	14.8	12.3
	#4	0.7	5.8	13.0	13.9	11.5
		0.6	6.7	13.7	14.8	12.3
		0.5	6.8	13.5	14.7	12.2

Table 1: Scores obtained with different rules and varying ω_{p}, for $\omega_{\mathrm{d}} = \omega_{\mathrm{h}} = 1.0$, playing safely.

Now, we vary the other parameters. For a full overview of the results, see [2]. It turns out that for the combination of hint rule #3 and discard rule #2, the best results can be obtained. In Figure 2 the results for varying ω-thresholds can be found, playing safely: no more risk is taken when playing cards after two errors have been made. We see (also in Table 1) that the best average score obtained is 15.4 when taking $\omega_{\mathrm{p}} = 0.6$, $\omega_{\mathrm{d}} = \omega_{\mathrm{h}} = 1.0$ and playing safely. Apparently, it is profitable to try and play a card once we are 60% sure that it will be correct (unless we have already made two errors in which case we require certainty), and only discard a card if it is certain that it is useless and give a hint if possible otherwise.

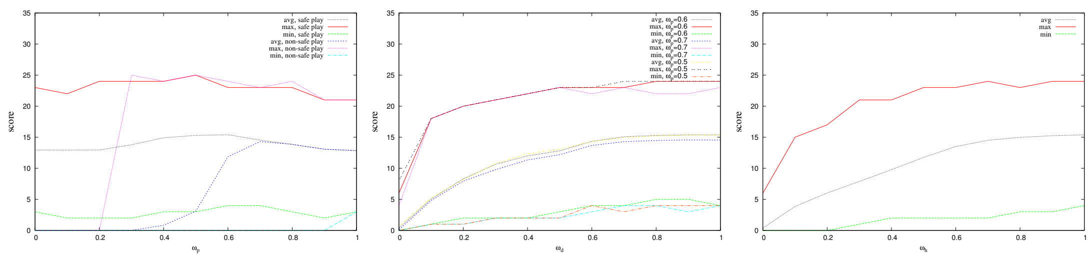

Figure 2: Scores obtained using hint rule #3 and discard rule #2, for varying ω-thresholds.

For the Monte Carlo strategy, the basis is well-known. In every turn, we try every action after which the game is played out by random players many times. Each of these random games is evaluated in some way, after which we choose to do the action which led to the best score.

In the implementation for HANABI, special care has to be taken in at least two situations. First, note that by trying to play a card and seeing how this turns out, a player could illegitimately obtain information on this card. Therefore, when trying to play a card in the Monte Carlo phase, the hand of the active player is shuffled through the deck and a new hand is dealt which is consistent with all hint information obtained so far. This way, the factual information on the cards is stored without allowing the agent to cheat. However, information on the exact hints that were given and the time at which these were given is lost, which possibly results in the agent not picking up implied hints, e.g., a hint pointing out a card actually meaning that it can successfully be played. This is in sharp contrast with the inner working of the strategies as described in [6]. Still, much progress could be booked here by better judging or even learning the probability distribution of the cards in the hand of the active player, incorporating the hints provided so far.

Second, a truly random player will end the game on three errors with high probability. To circumvent this, we choose to not end the game after three errors have been made in the play-out phase. Moreover, we prohibit the random player from playing a card of which it is certain that it cannot be played based on the hints received so far. If the random player knows that none of the cards in his/her hand can be played, he/she will always randomly discard a card or give a hint. If he/she knows that only some of his/her cards are useless, he/she may also randomly try to play one of the other cards.

Contrary to the standard implementation of Monte Carlo, we do not evaluate each of the play-outs on the final score obtained. Instead, we register for the next D turns the amount of new points obtained as well as the amount of errors made with respect to the current turn in each play-out. For each new point we administer a $+1$ and for the k-th error we administer a score of $-k$. We then compute a weighted sum over the scores for each of the D turns, counting later turns with (linear) higher weight, after which we pick the action with the highest average score among the play-outs. Note that taking the value $D = 1$ would result in greedy play.

Furthermore, it turns out to be profitable to not let the random player choose an action uniformly at random (indeed, such a player performs hardly better than a random player). If the random player decides to play a card, in the style of MCTS [4], he/she favors playing a card on which he/she has more information over playing a card on which he/she has less information. It seems a challenge not to incorporate too much game knowledge, certainly if it can be avoided.

In order to test the performance of the Monte Carlo player, we let three players using this strategy play 100 different starting configurations. A test run of this kind takes approximately thirty minutes on a computer with an Intel i7 2.3GHz core and 6 GB of RAM, when using 500 play-outs.

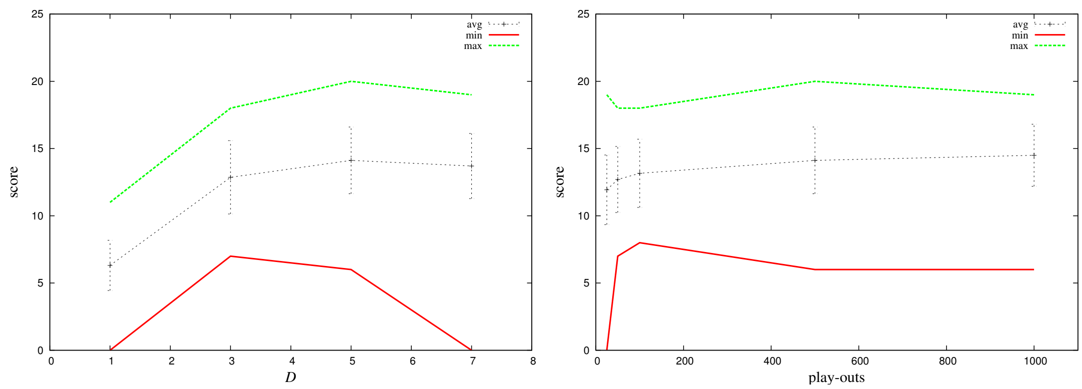

Figure 3: Scores obtained using 500 play-outs and varying D, and using $D = 5$ and varying number of play-outs, both with standard deviation.

In the experiments, the results of which can be seen in Figure 3, we have varied the amount of turns D that the Monte Carlo player takes into account and the number of play-outs allowed per different action. Surprisingly, the value of D taken, when not being too small, does not fundamentally affect the obtained score. Furthermore, while raising the amount of allowed play-outs does seems to somewhat improve the average score obtained, the differences are relatively small.

In fact, we note that it is not unlikely that these differences are to be contributed for a great part to random chance. Indeed, the standard deviation in the scores obtained seems to be quite high, with scores as low as 6 (or sometimes even 0) and as high as 20 being observed with an average around 14 points; the highest average score obtained is 14.5, with 1000 play-outs. We also see this large range in the scores obtained by the rule-based strategy discussed earlier in this section. It appears likely that the result of a game is heavily dependent on the starting configuration, and it could be worthwhile for the algorithm(s) to somehow adapt to this. As discussed in the previous section and in [1], it is sometimes impossible to score a perfect game with 25 points, in which case it may already be difficult to obtain a final score of more than 15. An easy example is a game where many 1s and/or 2s arrives late in the deck.

A provisional conclusion could be that, with the current settings, the rule-based players slightly outperform the Monte Carlo players, considering the one-point difference between the average scores obtained. The results from [6] for the two-player version, using online learning, even seem to be a little better than for the three-player version, although it could be that the two-player version allows for higher final scores.

5 Conclusions and Further Research

In this paper we have examined some interesting aspects of the cooperative card game HANABI, that has the special property that players can only see the cards in the hands of the other players — but not in their own. Hints provide partial information about the game states.

Even simplified versions of the game, like SINGLEHANABI, give rise to complicated combinatorial questions and corresponding formulas. Furthermore, we have shown that different game playing strategies offer promising results. In particular, Monte Carlo techniques require little game knowledge, but are capable of delivering high quality competitive players. One issue is the problem of the amount of information that is used during the play-outs. The rule-based players perform a little bit better, at the cost of incorporating more game knowledge.

As further research we first mention the quest for a proof of the result from Section 3 using the principle of inclusion and exclusion, perhaps also providing links to more general theorems. It is also of interest to find (substantial) classes of games that are unplayable. For the Monte Carlo players, we like to combine the presented method with techniques as mentioned in [4], like UCT, meanwhile somehow learning knowledge. And finally, there is much potential in research regarding the hints system.

Acknowledgements

The authors would like to thank Anne Hommelberg and Hendrik Jan Hoogeboom, as well as the anonymous contributors from [9].

References

[1] J.-F. Baffier, M.-K. Chiu, Y. Diez, M. Korman, V. Mitsou, A. van Renssen, M. Roeloffzen and Y. Uno, Hanabi is NP-complete, even for cheaters who look at their cards, in: Proceedings of 8th International Conference on Fun with Algorithms (FUN 2016), Leibniz International Proceedings in Informatics (LIPIcs) 49, pp. 4:1–4:17, 2016.

[2] M.J.H. van den Bergh, Hanabi, a cooperative game of fireworks, Bachelor thesis, Leiden University, www.math.leidenuniv.nl/scripties/BSC-vandenBergh.pdf, 2015. 1899848

[3] BoardGameGeek, website, www.boardgamegeek.com [retrieved 24.8.2016].

[4] 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 Transactions on Computational Intelligence and AI in Games 4, 1–43, 2012.

[5] H. van Ditmarsch and B. Kooi, One hundred prisoners and a light bulb, Springer, 2015.

[6] H. Osawa, Solving Hanabi: Estimating hands by opponent's actions in cooperative game with incomplete information, in: Proceedings of the Workshop at the Twenty-Ninth AAAI Conference on Artificial Intelligence: Computer Poker and Imperfect Information, pp. 37–43, 2015.

[7] R & R Games, website, www.rnrgames.com [retrieved 24.8.2016].

[8] S. Ruiz, An algebraic identity leading to Wilson's theorem, The Mathematical Gazette 80, 579–582, 1996.

[9] StackExchange, website, http://math.stackexchange.com/questions/1215764/sequences-that-contain-subsequence-1-2-3 [retrieved 24.8.2016].