Universiteit Leiden

Opleiding Informatica

Hanabi

A co-operative game of fireworks

Name: Mark van den Bergh

Date: 10/07/2015

Supervisors: dr. W.A. Kosters (LIACS)
dr. F.M. Spieksma (MI)

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Hanabi
A co-operative game of fireworks

Mark van den Bergh

July 10, 2015

Supervisors:
dr. W.A. Kosters
dr. F.M. Spieksma

Leiden Institute of Advanced Computer Science
Mathematisch Instituut
Universiteit Leiden

Abstract

Hanabi is a co-operative card game for two to five players, in which every player can see the contents of the other players' hands, but not of their own. By the exchange of hints, a player can obtain information about the cards in his or her hand.

The thesis consists of two main parts. In the first part, we study the notion of playability. Not every initial configuration of the game can result in a maximum score even if playing perfectly. By employing combinatorics, we derive a formula with which the amount of the initial configurations which can be finished perfectly can be calculated for a simplification of the original game. We also propose an approach using dynamic programming to perform these calculations for slightly more complicated versions of the game. In the second part, we test a variety of strategies in search of good strategies for the original game. We discover that some simple rules give promising results, but that not all strategies which seem good intuitively indeed result in high scores.

Contents

1 Introduction

Hanabi, Japanese for “fire flower” or “fireworks”, is a co-operative card game for two to five players. In contrast to many other games where every player tries to outperform the others in an attempt to secure victory, in Hanabi the players are required to work together and combine efforts to achieve the best possible result. This gives an entirely different dynamics to the game. An interesting feature of Hanabi is that players keep their cards in their hands with the card backs facing themselves. Hence, every player can see the value and suit of the cards in all other players’ hands, but not in their own. Indeed, viewing a game being played is quite odd at first.

In Section 2, after introducing some notation for sequences, we will start with a concise informal description of the rules of the game. Subsequently, we will formalise these rules in order to be able to construct rigid proofs. In Section 3, we will then look at the question of playability: for a given starting configuration of the game, it turns out that it might be impossible to obtain a perfect score. This situation can be compared to that in Klondike Solitaire, in which not all permutations of the stack are winnable [1]. The question of which configurations in Hanabi are playable turns out to give an interesting combinatorial problem, for which Theorem 3.25 is the main result.

Next, in Section 4, we will look at the one question which is interesting for every game: given a random initial configuration, what is a good or even optimal strategy? To answer this question, we will distinguish between two optimality criteria and run simulations to assess the quality of several strategies. Finally, in Section 5, the main results and conclusions of the thesis will be listed, as well as some questions which might give interesting directions for future research.

This thesis was written as part of the bachelor programmes of Mathematics and Computer Science at Leiden University, under the supervision of Floske Spieksma (MI) and Walter Kosters (LIACS).

2 Definitions and examples

In this section, we will provide an overview of the game of Hanabi in a series of definitions and examples. Before we delve into the game itself, however, we start with the introduction of some notation.

2.1 Preliminaries

First of all, we will use the following convention for the notation of natural numbers.

Notation 2.1. For the set of natural numbers, we write

$$\mathbb{N} = \{1, 2, 3, \ldots\} \quad \text{and} \quad \mathbb{N}_0 = \{0, 1, 2, 3, \ldots\}.$$

In the game of Hanabi, we deal with a stack of cards, which can be represented as a finite sequence in $\mathbb{Z}$ or $\mathbb{Z} \times \mathbb{Z}$. Therefore, it is useful to introduce some notation regarding these sequences.

Notation 2.2. Let $x = (x_i)_{i=1}^N$ and $y = (y_i)_{i=1}^M$ with $N, M \in \mathbb{N}$ be finite sequences. We write the concatenation of x and y as

$$xy = (x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_M).$$

Notation 2.3. Let $x = (x_i)_{i=1}^N$ be a finite sequence with $x_i \in \mathbb{Z}$. We write

$$x - 1 = (x_1 - 1, x_2 - 1, \ldots, x_N - 1).$$

Notation 2.4. Let $x = (x_i)_{i=1}^N$ be a finite sequence. We write

$$\{x\} = \{x_i \mid i = 1, \ldots, N\}$$

for the multiset consisting of the sequence entries. If for some multiset S we have $S \subseteq \{x\}$, we write $S \subseteq x$.

Notation 2.5. Let $x = (x_i)_{i=1}^N$ be a finite sequence. By $x \backslash x_i$, we denote x with the i-th element removed:

$$x \backslash x_i = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N).$$

Furthermore, in Section 3, we will make extensive use of binomial and multinomial coefficients. In the proofs in this section, the rules of Pascal's triangle are often necessary to rewrite our expressions. We state them here as lemmas, the proofs of which can be found in Theorem 1.5.1 in [2].

Lemma 2.6. *For $n, k \in \mathbb{N}_0$ with $n \geqslant k$, we have*

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

Lemma 2.7. *For $n, k_1, \ldots, k_m \in \mathbb{N}_0$ with $\sum_{i=1}^m k_i = n$, we have*

$$\binom{n}{k_1, \ldots, k_m} = \binom{n-1}{k_1 - 1, k_2, \ldots, k_m} + \binom{n-1}{k_1, k_2 - 1, \ldots, k_m} + \ldots + \binom{n-1}{k_1, k_2, \ldots, k_m - 1}.$$

2.2 Rules of classic Hanabi

Hanabi is a co-operative card game in which the players need to work together to achieve the highest possible score. What follows is a concise description of the rules of the game. For a more detailed version of the rules, consult [3].

In the base game, which we will denote by *classic Hanabi*, we have a stack of 50 cards. Every card has both a value and a suit, where we distinguish between 5 different values and suits. Hence, there are $5 \cdot 5 = 25$ unique cards in the stack. For every suit, we have three cards with value 1, two cards with values 2, 3 and 4 and one card with value 5, giving $3 + 2 + 2 + 2 + 1 = 10$ cards per suit and thus 50 cards in the stack in total.

The classic game is played with 2 to 5 players. At the start of the game, every player is dealt an opening hand, which consists of 5 cards if playing with 2 or 3 players and 4 cards if having 4 or 5 players. Each player may view the cards in all other player's hands. However, a player does not know the contents of his or her own hand. The goal of the game is to form five stacks, one of every suit, every stack being built up from 1 to 5.

During the game, players take turns in clockwise order. On a turn, a player must perform one of the following three actions:

1. give a hint,
2. discard a card,
3. play a card.

To give a hint, a player must spend one of the available hint tokens, of which 8 are given at the start. If no hint tokens are available, a player cannot perform this action. To give a hint, a player indicates all cards of any one number or any one suit in another player's hand.

Example 2.8. John's hand consists of a red 1, a green 4, a green 5, a white 1 and a white 4. On her turn, Mary may spend a hint token to indicate the 5 in John's hand, for example. She could also point out the two green cards, or the two 1s. She is not allowed to point out a single 1, however. A hint must give complete information about a single suit or number.

To discard a card, a player simply picks a card from his or her hand and moves it to the discard pile. A new card is drawn from the stack to replenish the player's hand and a hint token is added to the stockpile. If all 8 hints are already available, a player cannot discard.

To play a card, a player picks a card from his or her hand and puts it face-up on the table. If the card played is a 1 of a suit of which there is not yet a stack on the table, it forms the beginning of a new stack. Otherwise, the card is added to the stacks of fireworks of the proper suit already on the table, if possible: stacks must be built up from 1 to 5 in that order and may not contain duplicate cards. If a stack is completed, i.e., if a 5 is succesfully played, a hint is added to the stockpile if possible. If the card cannot be played in either of these fashions, it is moved to the discard pile and an error is noted. In any case, the player draws a new card to replenish his or her hand.

The game ends immediately if a third error is made, resulting in a score of 0. The game also ends immediately if the fifth stack is completed, resulting in a score of 25 points. If the bottom card of the stack is drawn, every player (including the one that drew the last card) may perform one more action. After that, one calculates the sum of the highest numbered card of every suit, resulting in a score between 0 and 25.

2.3 Initial configurations

To be able to reason in a mathematically robust way about the game of Hanabi, we will introduce a series of definitions. These definitions will allow us to generalise the game to an arbitrary number of suits, players, etc. as well. We will start by defining the initial configuration of a game, which we will take as the definition of a game itself.

Definition 2.9. An *initial configuration* or simply a *game* of Hanabi is defined as a 7-tuple $H = (n, k, p, h, S_0, t_0, f_0)$. Denoting $\text{Cards}(H) = \{1, \ldots, n\} \times \{1, \ldots, k\}$, we interpret the parameters in the following way:

(1) $n \in \mathbb{N}$ is the amount of available card *values*,

(2) $k \in \mathbb{N}$ is the amount of available *suits*,

(3) $p \in \mathbb{N}$ is the amount of *players*,

(4) $h \in \mathbb{N}$ is the *hand size* of every player,

(5) $S_0 = (s_i)_{i=1}^{N}$ is an ordered sequence of elements $s_i \in \text{Cards}(H)$ with $N \in \mathbb{N} \cup \{\infty\}$ such that $p \cdot h \leqslant N$ forming the *initial stack*,

(6) $t_0 \in \mathbb{N}_0 \cup \{\infty\}$ is the amount of *hints* initially available and

(7) $f_0 \in \mathbb{N} \cup \{\infty\}$ is the amount of *errors* after which the game ends.

In pratice, when choosing $t_0 = \infty$, we will instead take a finite upper bound on the amount of hints that can be given which is large enough to be able to give each and every possible hint at any stage in the game. By using this bound, we guarantee that a playout of a game of Hanabi remains finite.

Example 2.10. For classic Hanabi as described in the previous paragraph, we have $H = (5, 5, p, h, S_0, 8, 3)$ with $p \in \{2, \ldots, 5\}$,

$$h = \begin{cases} 5, & \text{if } p = 2 \text{ or } p = 3, \\ 4, & \text{if } p = 4 \text{ or } p = 5 \end{cases}$$

and $S_0 = (s_i)_{i=1}^{50}$ with $s_i = (x_i, y_i)$ being such that $\#\{s = (1, y)\} = 3$ for every $y = 1, \ldots, 5$, $\#\{s = (x, y)\} = 2$ for every $x = 2, 3, 4$, $y = 1, \ldots, 5$ and $\#\{s = (5, y)\} = 1$ for every $y = 1, \ldots, 5$.

Example 2.11. The boxed version of Hanabi features a sixth 'rainbow' suit as an extension of the classic game. In the original German version of the game, the sixth colour is indeed a full-fledged sixth suit, leading to the game described by $H = (5, 6, p, h, S, 8, 3)$ with p and h as in Example 2.10 and $S_0 = (s_i)_{i=1}^{60}$

with $s_i = (x_i, y_i)$ being such that $\#\{s = (1, y)\} = 3$ for every $y = 1, \ldots, 6$, $\#\{s = (x, y)\} = 2$ for every $x = 2, 3, 4$, $y = 1, \ldots, 6$ and $\#\{s = (5, y)\} = 1$ for every $y = 1, \ldots, 6$.

Example 2.12. In the Dutch version of the game, the rainbow suit adds fewer cards to the stack. Here, we obtain H as in Example 2.11, but now $S_0 = (s_i)_{i=1}^{55}$ with $s_i = (x_i, y_i)$ being such that $\#\{s = (1, y)\} = 3$ for every $y = 1, \ldots, 5$, $\#\{s = (x, y)\} = 2$ for every $x = 2, 3, 4$, $y = 1, \ldots, 5$, $\#\{s = (5, y)\} = 1$ for every $y = 1, \ldots, 5$ and $\#\{s = (x, 6)\} = 1$ for every $x = 1, \ldots, 5$.

It is useful to define the different areas in which cards can be played during the course of the game.

Definition 2.13. Given a game of Hanabi $H = (n, k, p, h, S_0, t_0, f_0)$ with $S_0 = (s_i)_{i=1}^{N}$, we define the following.

(1) The remaining *stack* is a sequence $S = (s_i)_{i=L}^{N}$ for some L with $1 \leqslant L \leqslant N$. We may also have $S = \varnothing$.

(2) The *discard pile* is a sequence $D = (d_i)_{i=1}^{M}$ with $d_i \in \text{Cards}(H)$ such that $M \leqslant N$.

(3) The *hand* P_j of player j, $j = 1, \ldots, p$, is a multiset containing elements of $\text{Cards}(H)$.

(4) The *field stacks* $F[i] = (f_{ij})_j$ are increasing sequences with elements in $\text{Cards}(H)$ for $i = 1, \ldots, k$.

At the start of the game, the discard pile and field stacks are all empty. Note furthermore that the cardinality of P_j will be h for most of the game, except for possibly after the last turn of a player (at which point it becomes $h - 1$, because a new card cannot be drawn if the stack is empty). Finally, the multiset consisting of $\{S\} \cup \{D\} \cup \bigcup_{i=1}^{p} P_i \cup \bigcup_{i=1}^{k} \{F[i]\}$ will be equal to the multiset $\{S_0\}$ containing the elements of the initial stack at any time during the game.

2.4 Process of play

We will now define the dynamic process that corresponds to the playing of a game of Hanabi. To do so, we first need the notion of knowledge. Formally, the knowledge about a card s_i is uniquely bound to the index i. However, to avoid cumbersome indices, we will describe the knowledge as being connected to a card $c \in \text{Cards}(H)$, even though some cards might occur multiple times in the game. As every card is still uniquely identifiable, this does not cause problems in practice. We also slightly abuse notation by using set operators when dealing with multisets.

Definition 2.14. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi and $c \in P_j$ a card in the hand of player j. The *knowledge* available about c is a subset $K_c \subseteq \text{Cards}(H)$, where $d \in K_c$ if player j believes that $c = d$ might hold. The *knowledge base* of player j is the multiset $\mathcal{K}_j = \{K_c\}_{c \in P_j}$.

For any game of Hanabi $H = (n, k, p, h, S_0, t_0, f_0)$, at the start of the game, we see that $K_c = \text{Cards}(H)$ for every $c \in P_j$, $j = 1, \ldots, p$ as the players do not have any information about their cards. To show the evolution of a player's knowledge due to hints given by another player, we consider the following example.

Example 2.15. Consider classic Hanabi H with $p = 2$ as in Example 2.10. Let $P_1 = \{(1,1),(4,2),(5,2),(1,3),(4,3)\}$ and $K_c = \text{Cards}(H)$ for every $c \in P_1$. On his turn, player 2 points out the two ones in the hand of player 1. After this hint, the knowledge base of player 1 becomes $\mathcal{K}_1 = \{A, B, B, A, B\}$ where $A = \{(x,y) \in \text{Cards}(H) \mid x = 1\}$ and $B = \{(x,y) \in \text{Cards}(H) \mid x \neq 1\}$.

The given definition of knowledge thus also allows us to formalise the giving of hints. This leads to the following description of the process of playing a game of Hanabi.

Algorithm 2.16 (Playing a game of Hanabi)**.** Given a game of Hanabi $H = (n, k, p, h, S_0, t_0, f_0)$ with $S_0 = (s_i)_{i=1}^N$, this algorithm describes the playing of the game:

1. Initialisation. We set $D := \varnothing$ and $F[i] := \varnothing$ for every $i = 1, \ldots, k$. Let $P_j := \{a_j, \ldots, a_{j+h-1}\}$ be the starting hand for every player $j = 1, \ldots, p$ and set $S := (s_i)_{i=1+ph}^N$. Let $C := 1$, $t := t_0$, $f := 0$, $M := 0$ and $R = \infty$.

2. Player C performs any one of the following actions:

   (a) Give a hint. If $t \geqslant 1$, pick $i \in \{1, \ldots, p\}$ and $j \in \{1, 2\}$. If $j = 1$, pick $a \in \{1, \ldots, n\}$ and set

   $$K_c = \begin{cases} K_c \backslash \{(x,y) \in \text{Cards}(H) \mid x \neq a\}, & \text{if } c = (a, y) \\ & \text{ for any } y \in \{1, \ldots, k\}, \\ K_c \backslash \{(x,y) \in \text{Cards}(H) \mid x = a\}, & \text{if } c \neq (a, y) \\ & \text{ for all } y \in \{1, \ldots, k\} \end{cases}$$

   for all $c \in P_i$. If $j = 2$, pick $a \in \{1, \ldots, k\}$ instead and set

   $$K_c = \begin{cases} K_c \backslash \{(x,y) \in \text{Cards}(H) \mid y \neq a\}, & \text{if } c = (x, a) \\ & \text{ for any } x \in \{1, \ldots, n\}, \\ K_c \backslash \{(x,y) \in \text{Cards}(H) \mid y = a\}, & \text{if } c \neq (x, a) \\ & \text{ for all } x \in \{1, \ldots, n\} \end{cases}$$

   for all $c \in P_i$. Set $t := t - 1$.

   (b) Discard a card. Pick $a \in P_C$. Define $P_C := (P_C \backslash a) \cup \{s_1\}$ and $\mathcal{K}_C := (\mathcal{K}_C \backslash K_a) \cup \text{Cards}(H)$. Set $S := S \backslash s_1$ and $D := (d_1, d_2, \ldots, d_M, a)$. Set $M := M + 1$. If $t < t_0$, set $t := t + 1$.

   (c) Play a card. Pick $a \in P_C$. Define $P_C := (P_C \backslash a) \cup \{s_1\}$ and $\mathcal{K}_C := (\mathcal{K}_C \backslash K_a) \cup \text{Cards}(H)$. Set $S := S \backslash s_1$. If $a = (x, 1)$ and $F[x] = \varnothing$, then $F[x] := (1)$. If $a = (x, y)$ and $F[x] = (1, \ldots, y-1)$, then $F[x] := (1, \ldots, y)$. If furthermore $y = n$ and $t < t_0$, then $t := t+1$. Otherwise, $D = (d_1, d_2, \ldots, d_M, a)$ and set $M := M+1$ and $f := f+1$. If $f = f_0$, set $R = -1$.

   After making a move, set $C := (C + 1) \mod p$. If $R > 0$, set $R := R - 1$.

3. If $R = -1$, STOP and return a score of 0. If $R = 0$, STOP and return a score of $\sum_{i=1}^{k} \max F[i]$. If $R > 0$ and $\max F[i] = n$ for all $i = 1, \dots, k$, STOP and return a score of $n \cdot k$. If $R = \infty$ and $S = \varnothing$, set $R := p$. Go to step 2.

In the algorithm, the variable R is used to keep track of the amount of turns remaining. Note that while the stack is not empty, we have no upper bound on the amount of turns left just yet, which we denote by $R = \infty$. Here, $\infty - 1 = \infty$.

Naturally, the way in which an action is chosen in step 2 defines the strategy of the players, which completely determines the outcome of the game. Note that the set of actions A from which a player can choose on any turn is always of size $(p-1) \cdot (n+k) + 2h$. If a player chooses to give a hint, there are n numbers and k suits to choose from and $p-1$ players to give a hint to; if a player chooses to play or discard a card, there are h cards to pick.

Definition 2.17. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi and A the set of available actions. A *decision rule* $x^t(D, F, \mathcal{K}) = (\omega_a)_{a \in A}$ is a probability distribution on A dependent on the current discard pile D, field stacks $F = (F[1], \dots, F[k])$, knowledge base $\mathcal{K} = (\mathcal{K}_1, \dots, \mathcal{K}_p)$ and turn t. A *strategy* is a sequence $x = (x^t)_t$ of decision rules. The *average value* $\overline{v}(x, H)$ of a strategy x is the average score achieved in H by playing x. The *maximum value* $v^*(x, H)$ is the maximum score achieved in H by playing the strategy x.

Example 2.18. Consider once again classic Hanabi as in Example 2.10. A possible decision rule is the random rule, in which every action has an equal possibility of being carried out every turn, i.e., $\omega_a = \frac{1}{10(p-1)+2h}$ for every $a \in A$, regardless of D, F, $\mathcal{K}$ and t. The random strategy x is formed by using the random rule every turn. The average value $\overline{v}(x, H)$ of the random strategy is close to 0 for any permutation of the stack as the probability of making three errors before ending the game in a different way is almost 1. Indeed, in the case of classic Hanabi where the probability of randomly playing a card succesfully can be approximated by $\frac{1}{5}$, chances of achieving the maximum score will be of order $\frac{1}{5^{25}}$.

Definition 2.19. Let H be a game of Hanabi. A decision rule $x^t(D, F, \mathcal{K}) = (\omega_a)_{a \in A}$ is *deterministic* if for all D, F and $\mathcal{K}$ there exists an $a^* \in A$ such that

$$\omega_a = \begin{cases} 1 & \text{if } a = a^*, \\ 0 & \text{otherwise.} \end{cases}$$

We may then note $x^t(D, F, \mathcal{K}) = a^*$. A strategy using only deterministic decision rules is called a *deterministic strategy.* The *value* $v(x, H)$ of a deterministic strategy x is the score achieved in H by playing x.

Definition 2.20. Let H be a game of Hanabi. A strategy $x = (x^t)_t$ is *stationary* if $x^t = x^*$ for any decision rule x^* and all $t \in \mathbb{N}_0$, i.e., if the decision rule used is independent of t.

3 Playability

Having set up the framework with which we will work, we now start on the first main question: given a game of Hanabi, can we achieve the maximum possible score if playing perfectly? We will first show that for answering this question, it is sufficient to regard an alternative of the original game with perfect information. We will then simplify the game further by reducing the amount of players and colours. For the simplest version of the game, some rigid calculations can be done.

3.1 Open games

For any given game of Hanabi, we wonder whether it is possible to score the maximum amount of points if we play optimally. To start our answer to this question, we introduce the notion of playability.

Definition 3.1. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi. H is *playable* if there exists a strategy x such that $v^*(x, H) = n \cdot k$. The strategy x *plays* H.

The following examples show that there exist non-playable games.

Example 3.2. Let $H = (5, 5, 2, 5, S_0, 8, 3)$ be 2-player classic Hanabi as defined in Example 2.10. Now, assume that S_0 is ordered in such a way that all fours and fives are on top of the stack, for example

$$S_0 = ((5, 1), (5, 2), \ldots, (5, 5), (4, 1), (4, 1), (4, 2), (4, 2), \ldots, (4, 5), (4, 5), \ldots).$$

At the start of the game, player 1 holds all fives and player 2 holds only fours, i.e., $P_1 = \{(5, 1), (5, 2), \ldots, (5, 5)\}$ and $P_2 = \{(4, 1), (4, 1), (4, 2), (4, 2), (4, 3)\}$. Note that to obtain a perfect score, we may never discard a 5: because there is only one in each suit, discarding would make it impossible to finish the stack in the corresponding suit. By the same reasoning, we may not discard both fours of the same suit. Therefore, the only course of action which might lead to a perfect score would be to discard fours from the hand of player 2, keeping at least one four of every suit in the game.

After some turns, this will yield $P_2 = D = \{(4, 1), (4, 2), \ldots, (4, 5)\}$. Now, note that at this point there is exactly one 4 left of every suit. However, after possibly expending all available hints, we must discard or play another card to move on. With both P_1 and P_2 now consisting of cards which cannot rightfully be discarded, we see that we cannot obtain a score of 25 points using this permutation of the stack.

Example 3.3. Again consider classic Hanabi $H = (5, 5, p, h, S_0, 8, 3)$, but now with an arbitrary number of players. Let S_0 be such that all ones are on the bottom of the stack, for example

$$S_0 = (\ldots, (1, 1), (1, 1), (1, 1), (1, 2), (1, 2), (1, 2), (1, 3), \ldots, (1, 5)).$$

Now, note that no card can succesfully be played until the first 1 is drawn from the stack. As such, the maximum amount of cards left in the stack when the first field stack becomes non-empty is 14. Whenever a card is played, a card is also drawn from the stack. Therefore, after at most 14 cards have been played to the field stacks, we find $S = \varnothing$.

After the stack has been emptied, every player including the one who drew the last card may execute one more action. In the optimal case that every player can play another card in this last round of actions, we can add p more cards to the field stacks. However, in classic Hanabi, $p \leqslant 5$. Therefore, when the game ends, at most 14+5=19 cards have been added to the field stacks and as such the attained score cannot be higher than 19 – it certainly cannot be 25.

These examples show that there are at least two distinct causes which can make a game unplayable. First, we may find that there is not enough space in player's hands to store crucial cards until they can be played. In other words, we might at some point be forced to discard a card which is actually needed later on. Second, we see that even if this situation does not arise, we might not have enough time to play enough cards, i.e., the game ends because of an empty stack before $n \cdot k$ cards were played.

Now, we would like to investigate what percentage of permutations of a given stack are actually playable. As the giving of hints and processing of information is a difficult aspect of the game, we will look at a version of Hanabi which features perfect information. If it is impossible to achieve a perfect score even if the players can see everything in the game, it will certainly be impossible to obtain the maximum score if we need to give hints and deduce what cards we have. Therefore, considering open games will certainly provide an upper bound on the percentage of playable configurations of a given stack. We start with some definitions and examples.

Definition 3.4. An *(initial configuration of an) open game* of Hanabi is defined as a 7-tuple $L = (n, k, p, h, S_0, t_0, f_0)$, where $t_0 \in \mathbb{N}_0 \cup \{\infty\}$ is now the amount of *passes* initially available: the action of giving a hint to a fellow player is replaced by the action of passing the turn to the next player. For $t_0 = \infty$, we use the same convention as introduced below Definition 2.9. The parameters n, k, p, h, S_0 and f_0 are also as in Definition 2.9. For a given game of Hanabi H, we write H_o for its *associated open game.*

Note that the areas of play are still the same as in Definition 2.13. Also, Algorithm 2.16 can be used almost unaltered to play a given open game L. We simply replace step 2a by “Pass: do nothing.”. Moreover, we no longer need the knowledge base of Definition 2.14 as we do not have hints in an open game. As such, we may also remove the operations on $\mathcal{K}_C$ in steps 2b and 2c. This change in available information is also reflected in the definitions of a decision rule and strategy in an open game.

Definition 3.5. Let $L = (n, k, p, h, S_0, t_0, f_0)$ be an open game of Hanabi and A the set of available actions. An *open decision rule* $x_o^t(D, F, S, \mathcal{P}) = a \in A$ is a function which gives an action in A dependent on the current discard pile D, field stacks $F = (F[1], \ldots, F[k])$, stack S, hand cards $\mathcal{P} = (P_1, \ldots, P_p)$ and turn t. An *open strategy* is a sequence $x_o = (x_o^t)_t$ of open decision rules. The *value* $v(x_o, H)$ is the score achieved in H by playing the strategy x_o. If $x_o^t = x_o^*$ for any open decision rule x_o^* and for all $t \in \mathbb{N}_0$, the strategy $x_o = (x_o^t)_t = (x_o^*)_t$ is called *stationary.*

Note that we now no longer need non-determinism to define our strategies because we have access to perfect information: we do not need to take chances, because we know exactly what will be the result of our actions. Now, playability of an open game of Hanabi is defined in the same way as playability of a regular game as in Definition 3.1. However, a decision rule and thus a strategy in an open game is not the same as a strategy in a regular game, as different amounts of information are available to determine the course of action. As mentioned earlier, if a regular game is playable, then certainly its associated open game is playable. In fact, the converse also holds. To obtain this result, we provide the following lemma.

Lemma 3.6. *Let H be a playable game of Hanabi. If H is playable, then there exists a deterministic strategy x which plays H.*

Proof. Because H is playable, there exists a finite sequence of actions $(a_i)_{i=1}^N$ for some $N \in \mathbb{N}_0$ such that performing these actions in order results in a score of $n \cdot k$. Define the decision rule x^t by $x^t(D, F, \mathcal{K}) = a_t$ for $t = 1, \ldots, N$, regardless of D, F and $\mathcal{K}$. Consider the strategy $x = (x^t)_t$. This strategy is obviously deterministic and has value $v(x, H) = n \cdot k$ by construction, hence x plays H. □

Theorem 3.7. *Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi. H is playable if and only if H_o is playable.*

Proof. First, assume that H is playable. By Lemma 3.6, there exists a deterministic strategy x such that $v(x, H) = n \cdot k$. Run the game using this strategy x in Algorithm 2.16. In every execution of step 2, let $x_o^t(D, F, S, \mathcal{P}) = x^t(D, F, \mathcal{K})$, where a hint is replaced by a pass. For any other 5-tuple D, F, S, $\mathcal{P}$ and t let $x_o^t(D, F, S, \mathcal{P})$ be a pass. Now, playing the open game H_o, the strategy $x_o = (x_o^t)_t$ will mimic the actions taken when playing H using x. As such, we will keep facing the same game states and achieve a perfect score.

Now suppose that H_o is playable. Let x be the random strategy as described in Example 2.18, i.e., $x^t(D, F, \mathcal{K}) = \frac{1}{(p-1)\cdot(n+k)+2h}$ for all D, F, $\mathcal{K}$ and t and for all $a \in A$. Because H_o is playable, there exists a sequence of actions which results in a score of $n \cdot k$. As the random strategy x may result in any sequence of actions, it may specifically result in the sequence giving a perfect score. Therefore, the maximum value $v^*(x, H)$ is $n \cdot k$, hence H is playable. □

3.2 Single-player Hanabi

We will now first narrow our analysis to that of single-player open games of Hanabi. In these games, the action of passing is obviously futile as the turn cannot be passed to another player.

Theorem 3.8. *Let $L = (n, k, 1, h, S_0, t_0, f_0)$ be a single-player open game of Hanabi. Let $L_0 = (n, k, 1, h, S_0, 0, f_0)$. If L is playable, then L_0 is playable.*

Proof. Let x_o be the strategy that plays L. Assume that $x_o^t(D, F, S, \mathcal{P})$ is a pass for some D, F, S, $\mathcal{P}$ and t^*. If this situation is met during the playing of the game, we will execute action 2a in the modified Algorithm 2.16, by which none of D, F, S and $\mathcal{P}$ is changed. Consequently, it is still the same player's turn while the state of the game has not changed, except for the turn number increasing by 1. Therefore, the strategy x'_o defined by

$$x'^t_o(D, F, S, \mathcal{P}) = \begin{cases} x^t_o(D, F, S, \mathcal{P}), & \text{if } t < t^*, \\ x^{t+1}_o(D, F, S, \mathcal{P}), & \text{if } t \geqslant t^* \end{cases}$$

will result in the same course of the game and thus the same perfect score of $n \cdot k$. We can eliminate all passes from the given strategy x in this fashion, noting that x cannot consist of infinitely many passes by the convention for $t_0 = \infty$. □

Corollary 3.9. *Let $L = (n, k, 1, h, S_0, t_0, f_0)$ be a single-player open game of Hanabi. Let $L_x = (n, k, 1, h, S_0, x, f_0)$. If L is playable, then L_x is playable for all $x \in \mathbb{N}_0$.*

In the single-player situation, it is thus sufficient to consider only games without passing. But even with this knowledge, it turns out to be difficult to construct an open strategy which always plays a playable open game. Naturally, a card in hand should be played when possible. However, determining which card to discard when facing a hand of cards which cannot be played is hard. We will return to this matter later on, but first our attention shifts to some special cases.

3.3 Playability of single-colour sequences

To further narrow down our search, we will consider games with only one suit. For such games L, the set of available cards is given by $\text{Cards}(L) = \{1, \ldots, n\} \times \{1\} \cong \{1, \ldots, n\}$. As such, the stack S can now be represented by a simple ordered sequence of integers. We will extend our definition of playability of games to these sequences.

Definition 3.10. Let $S = (s_i)_{i=1}^N$, $s_i \in \mathbb{Z}$ be an ordered sequence. Then S is (k, m)-*playable* if there exist an open game $L = (n, 1, 1, m, S, 0, 0)$ and open strategy x_o such that $v(x_o, L) = k$.

Example 3.11. Consider the sequence $S = (4, 3, 2, 1, 1, 1, 1)$. We might wonder whether S is $(4, 4)$-playable, i.e., if our single player has a hand size of 4, can he achieve a score of 4? The starting hand will be $P = \{4, 3, 2, 1\}$ and the stack will be $S = (1, 1, 1)$ at the start of the first turn. On this first turn, we play the 1 and draw a card to replenish our hand, leading to $P = \{4, 3, 2, 1\}$ and $S = (1, 1)$. We can now play a 2 and after drawing a card we then have $P = \{4, 3, 1, 1\}$ and $S = (1)$. Playing the 3 then yields $P = \{4, 1, 1, 1\}$ and $S = \varnothing$. As the stack is now empty, we have one more turn left, in which we can play the 4 to complete the goal of scoring 4 points. Therefore, S is $(4, 4)$-playable.

However, S is not $(4, 1)$-playable. In this scenario, the starting hand and stack will be $P = \{4\}$ and $S = (3, 2, 1, 1, 1, 1)$ respectively. On our first turn, we are forced to discard the 4. Unfortunately, we see that we will not encounter another 4 in the rest of the stack and therefore cannot obtain a score of 4 points.

Example 3.12. Let $S = (1, 2, 3, 4)$. In contrast with the stack in the previous example, this stack is $(4, 1)$-playable. At the start of the first turn, we have $P = \{1\}$ and $S = (2, 3, 4)$. We can thus play the 1, after which we draw the 2, which we can again play, etc. Ending up with an empty stack and just enough time to put down the 4, we indeed see that S is $(4,1)$-playable.

Now we wonder whether S is also $(4,4)$-playable. The initial hand will then be $P = \{1,2,3,4\}$ and the initial stack is $S = \varnothing$. As the stack is already empty at the start, we only have one turn to play a card after which the game ends immediately. Therefore, no score higher than 1 can be reached, thus S is not $(4,4)$-playable.

These two examples show that no simple ordering exists in the second argument: a (k,m)-playable sequence is not necessarily (k,n)-playable for all $n < m$ or all $n > m$. Such an ordering does exist in the first argument, however, which together with three simple other properties make up the following theorem.

Theorem 3.13. *Let $S = (s_i)_i^N$, $s_i \in \mathbb{Z}$ be an ordered sequence.*

1. *If S is (k,m)-playable, then S is (l,m)-playable for all $0 \leqslant l < k$.*
2. *S is $(k,1)$-playable if and only if S contains the sequence $(1,\dots,k)$ as a subsequence.*
3. *If $\{1,\dots,k\} \subsetneq \{S\}$, then S is not (k,m)-playable for all $m \in \mathbb{N}_0$.*
4. *If $N - m + 1 < k$, then S is not (k,m)-playable.*

Proof. 1. If S is (k,m)-playable, there exist an open strategy x_o and an open game $L = (n,1,1,m,S,0,0)$ such that $v(x_o,L) = k$. Now, define $x'_o = (x'^t_o)_t$ by

$$x'^t_o(D,F,S,\mathcal{P}) = \begin{cases} x^t_o(D,F,S,\mathcal{P}) & \text{if } |F| < l, \\ \text{discard random card} & \text{if } |F| = l. \end{cases}$$

By playing x'_o, we imitate x_o until we obtain a score of l points, after which no more cards are played and thus no more points are obtained. As such, x'_o scores exactly l points, which proves the (l,m)-playability of S.

2. Let P_t be the player's hand on turn t, $t \geqslant 1$. Note that the hand size of 1 and the lack of passing in a single-player game ensure that $P_t = \{s_t\}$ for all $t = 1,\dots,N$: every turn, the one hand card is played or discarded and subsequently replaced by the top card of the stack.

   Now, if S is $(k,1)$-playable, we must have $s_{t_1} = 1$, $s_{t_2} = 2$, ..., $s_{t_k} = k$ for some $t_1 \leqslant t_2 \leqslant \dots \leqslant t_k$, as a card with value $i > 1$ can only be played if a card with value $i-1$ was played before. As such, we see that $(1,\dots,k)$ is a subsequence of S.

   Conversely, assume that $(1,\dots,k)$ is a subsequence of S. Now, let $L = (n,1,1,m,S,0,0)$ with $n = \max(S)$ be an open game. By assumption, $s_{t_1} = 1$, $s_{t_2} = 2$, ..., $s_{t_k} = k$ for some $t_1 \leqslant t_2 \leqslant \dots \leqslant t_k$. Define the stationary open strategy x_o by

$$x_o(D,F,S,\mathcal{P}) = \begin{cases} \text{play card} & \text{if } P_t = \{s_t\}, |F| = s_t - 1 < k, \\ \text{discard card} & \text{otherwise.} \end{cases}$$

   This strategy succesfully plays cards onto the field stack until a score of k points is reached.

3. The multiset $\{S\} \cup \{D\} \cup \{P_1\} \cup \{F[1]\}$ is equal to the multiset of the initial stack at any point during the game. If k points need to be scored in an open game $L = (n, 1, 1, m, S, 0, 0)$, the field stack must be equal to $F[1] = \{1, \dots, k\}$ at some point in the game. However, as $\{1, \dots, k\} \nsubseteq \{S\}$, we have $\{1, \dots, k\} \nsubseteq \{F[1]\}$ at any time during the game.

4. Let S_t denote the stack at the start of turn t, $t \geqslant 1$. By Algorithm 2.16, $S_1 = (s_i)_{i=m+1}^N$ with $|S_1| = N - m$. As we play or discard a card on every turn in a single-player open game, we find that $|S_t| = N - m - t + 1$ for every $t \geqslant 1$. Then, for $t = N - m + 1$, we find that $|S_t| = 0$ and as such $S_t = \varnothing$. By the rules of Hanabi, we can thus have a maximum of $N-m+1$ turns and therefore play a maximum of $N-m+1$ cards. To score k points, we need to play exactly k cards. However, $k > N - m + 1$ by assumption.

□

Part 2 of the given theorem gives rise to a combinatorical approach to calculating the amount of $(k, 1)$-playable sequences, which we will now elaborate upon.

3.4 The case $m = 1$

In this section, we will calculate the amount of $(k, 1)$ playable sequences which can be constructed for a given number of 1s, 2s, etc. For example, if we are given two 1s, three 2s and one 3, how many $(3, 1)$-playable sequences can we construct using these numbers? Theorem 3.13.2 states that this is exactly the amount of sequences we can construct using the given numbers which contain the sequence $(1, 2, 3)$ as a subsequence.

Definition 3.14. Let $x_1, x_2, \dots, x_k \in \mathbb{N}_0$ with $\sum_{i=1}^k x_i = N$. We then denote $\mathrm{Seq}(x_1, \dots, x_n)$ to be the set of all sequences for which $\#\{x \in S \mid x = i\} = x_i$ for all $i = 1, \dots, k$. The set $\mathrm{Pl}(x_1, \dots, x_k) \subseteq \mathrm{Seq}(x_1, \dots, x_k)$ is defined to be the subset containing exactly all $(k, 1)$-playable sequences.

Notation 3.15. We write

$$\# \mathrm{Seq}(x_1, \dots, x_k) = T(x_1, \dots, x_k),$$

$$\# \mathrm{Pl}(x_1, \dots, x_k) = \chi(x_1, \dots, x_k),$$

$$\#(\mathrm{Seq}(x_1, \dots, x_k) \backslash \mathrm{Pl}(x_1, \dots, x_k)) = \psi(x_1, \dots, x_k).$$

Note that we have

$$T(x_1, \dots, x_k) = \binom{N}{x_1, \dots, x_k} = \frac{N!}{x_1! \cdots x_k!} \tag{3.4.1}$$

for any given x_i by [4]. Calculating the number $\chi(x_1, \dots, x_k)$ turns out to be less trivial, however.

Example 3.16. We calculate $\chi(2, 2, 1)$ by inspection: how many sequences are there consisting of two 1s, two 2s and one 3 which contain $(1, 2, 3)$ as a subsequence? First of all, note that there are $T(2, 2, 1) = \binom{5}{2,2,1} = 30$ sequences containing the right number of digits in total. By inspecting every one of these sequences, we see that only 11223, 12123, 12213, 21123, 21213, 11232, 12132, 12231, 21231, 12312 and 12321 contain $(1, 2, 3)$ as a subsequence, resulting in $\chi(2, 2, 1) = 11$.

With the total number of sequences $T(x_1, \ldots, x_k)$ growing large very fast for increasing x_i, simply inspecting all possible sequences rapidly becomes an infeasible approach. In the next section, a recursive method of determining whether a sequence is (k, m)-playable for arbitrary m is discussed. For the case $m = 1$, we can derive an expression for $\chi(x_1, \ldots, x_k)$. We will do this in a number of steps, increasing k by one at a time.

Theorem 3.17. *Let $x_1, x_2 \in \mathbb{N}_0$ be such that $x_1 + x_2 = N$. Then*

$$\chi(x_1, x_2) = \binom{N}{x_1} - 1.$$

Proof. By Eq. 3.4.1,

$$T(x_1, x_2) = \binom{N}{x_1, x_2} = \frac{N!}{x_1! x_2!} = \binom{N}{x_1}.$$

Note that a sequence in $\mathrm{Seq}(x_1, x_2)$ is $(2, 1)$-playable if there is at least one 1 which is followed by a 2 on any later position. The only sequence which does not contain such a 1 is the sequence in which all 2s occur before all 1s, hence

$$\underbrace{22\cdots2}_{x_2\text{digits}}\underbrace{11\cdots1}_{x_1\text{digits}}.$$

We thus find $\psi(x_1, x_2) = 1$, which completes the proof. □

Theorem 3.18. *Let $x_1, x_2, x_3 \in \mathbb{N}_0$ be such that $x_1 + x_2 + x_3 = N$. Then*

$$\chi(x_1, x_2, x_3) = \binom{N}{x_1, x_2, x_3} - 2^N + \sum_{i=0}^{x_1-1} \binom{N}{i} + \sum_{i=0}^{x_2-1} \binom{N}{i} + \sum_{i=0}^{x_3-1} \binom{N}{i}.$$

Proof. We give a direct combinatorial proof. By Eq. 3.4.1,

$$T(x_1, x_2, x_3) = \binom{N}{x_1, x_2, x_3}.$$

We will determine $\psi(x_1, x_2, x_3)$, from which $\chi(x_1, x_2, x_3)$ will then readily follow. First, note that every sequence $s \in \mathrm{Seq}(x_1, x_2, x_3) \setminus \mathrm{Pl}(x_1, x_2, x_3)$ can be divided into three parts P, Q and R, of which P does not contain any 1s, Q does not contain any 2s and R does not contain any 3s, i.e.,

$$s = \underbrace{[\text{2s and 3s only}]}_{P}\underbrace{[\text{1s and 3s only}]}_{Q}\underbrace{[\text{1s and 2s only}]}_{R}.$$

Note that P, Q and R may be empty. Indeed, let P be the part until the first 1 (if any, otherwise we are done), Q the part after P until the first 2 after the first 1 (if this exists) and R the rest. By construction, P does not contain any 1s and Q does not contain any 2s. If R would contain a 3, we found a subsequence

1-2-3, hence R does not contain a 3. Conversely, it is easy to see that a sequence of the form PQR does not contain a subsequence 1-2-3.

Now, a sequence in $\mathrm{Seq}(x_1, x_2, x_3) \setminus \mathrm{Pl}(x_1, x_2, x_3)$ is uniquely determined if we know the positions of all 3s and the positions of all 1s after these 3s. For example, given the sequence $xx3x3x33xxx3x1x1xx1x$, we first replace all x's to the left of the last 3 by 1s, starting from the rightmost one, until the number of 1s in the sequence equals x_1. By construction, the number of 3s is already equal to x_3 so that the replacement of all remaining x's will then lead to a sequence containing x_2 2s. Note that the resulting sequence is of the form PQR as described above and as such does not contain 1-2-3 as a subsequence. Furthermore, every initial placement of the 3s and 1s to the right of the 3s uniquely determines such a sequence and every sequence of the desired type can be created in this way.

In total, there are

$$\binom{N}{x_3} + \binom{N}{x_3+1} + \ldots + \binom{N}{x_3+x_1} = \sum_{i=0}^{x_1} \binom{N}{x_3+i}$$

choices for the initial placement of the 3s and 1s, as we need to place all x_3 of the 3s and only the 1s to the right of these 3s (which can be any amount between 0 and x_1 inclusive). However, note that the method proposed does not always work: if we have placed i 1s and are left with fewer x's to the left of the rightmost 3 than $x_1 - i$, there is not enough space to place the remaining 1s. To construct such a sequence, choose i, out of the N positions, $0 \leqslant i < x_1$. Now fill, starting from the left, x_3 non-selected positions with a 3 and the chosen positions to the right of the last 3 with a 1. This produces exactly the sequences that canot be completed in the indicated way. As $\binom{N}{i}$ selections can be made for every choice of i, this occurs for exactly

$$\binom{N}{0} + \binom{N}{1} + \ldots + \binom{N}{x_1-1} = \sum_{i=0}^{x_1-1} \binom{N}{i}$$

sequences in total. We thus find

$$\psi(x_1, x_2, x_3) = \sum_{i=0}^{x_1} \binom{N}{x_3+i} - \sum_{i=0}^{x_1-1} \binom{N}{i}.$$

Rewriting gives

$$\begin{aligned}
\sum_{i=0}^{x_1} \binom{N}{x_3+i} - \sum_{i=0}^{x_1-1} \binom{N}{i} &= \sum_{i=x_3}^{x_1+x_3} \binom{N}{i} - \sum_{i=0}^{x_1-1} \binom{N}{i} \\
&= 2^N - \sum_{i=0}^{x_3-1} \binom{N}{i} - \sum_{i=x_1+x_3+1}^{N} \binom{N}{i} - \sum_{i=0}^{x_1-1} \binom{N}{i} \\
&= 2^N - \sum_{i=0}^{x_3-1} \binom{N}{i} - \sum_{i=0}^{N-x_1-x_3-1} \binom{N}{i} - \sum_{i=0}^{x_1-1} \binom{N}{i} \\
&= 2^N - \sum_{i=0}^{x_3-1} \binom{N}{i} - \sum_{i=0}^{x_2-1} \binom{N}{i} - \sum_{i=0}^{x_1-1} \binom{N}{i}.
\end{aligned}$$

It follows that

$$\begin{aligned}\chi(x_1,x_2,x_3) &= T(x_1,x_2,x_3) - \psi(x_1,x_2,x_3)\\ &= \binom{N}{x_1,x_2,x_3} - 2^N + \sum_{i=0}^{x_1-1}\binom{N}{i} + \sum_{i=0}^{x_2-1}\binom{N}{i} + \sum_{i=0}^{x_3-1}\binom{N}{i}.\end{aligned}$$

□

Example 3.19. We calculate $\chi(3,3,2)$ using Theorem 3.18, i.e., how many sequences are there consisting of three 1s, three 2s and two 3s which contain $(1,2,3)$ as a subsequence? This turns out to be

$$\chi(3,3,2) = \binom{8}{3,3,2} - 2^8 + \sum_{i=0}^{2}\binom{8}{i} + \sum_{i=0}^{2}\binom{8}{i} + \sum_{i=0}^{1}\binom{8}{i} = 387.$$

By Eq. 3.4.1, the total number of sequences containing the given amounts of digits is $T(3,3,2) = \binom{8}{3,3,2} = 560$. We thus find that the fraction of playable sequences satisfying the given conditions is $\frac{387}{560} \approx 0.69$. In other words, if we shuffle the deck consisting of the specified cards randomly, the probability of the resulting game being playable by a single player with one card in hand is approximately 0.69.

In general, for the x_i large, Theorem 3.18 gives the following results on the fraction of playable sequences in $\mathrm{Seq}(x_1,x_2,x_3)$.

Corollary 3.20. *1.* $\lim_{N\to\infty} \frac{\chi(1,1,N)}{T(1,1,N)} = \frac{1}{2}$,

2. $\lim_{N\to\infty} \frac{\chi(1,N,N)}{T(1,N,N)} = 1$,

3. $\lim_{N\to\infty} \frac{\chi(N,N,N)}{T(N,N,N)} = 1$.

Intuïtively, these statements are clear. If there is a large supply of 3s, it is almost sure that there is at least one 3 both before and after the one 1 and 2. Therefore, the only scenario in which a sequence does not contain the subsequence 1-2-3 is the case in which the 2 appears before the 1, which occurs with probability $\frac{1}{2}$. If there is also a near-infinite supply of 2s, this implies that there are 2s and 3s both before and after the one 1 in any order. As such, the probability of a sequence not being playable tends to zero. Naturally, the same holds for sequences containing a large amount of every of the three numbers.

Now, note that the formula of Theorem 3.18 is symmetric in its three arguments: if we swap for example the amount of given 1s and 2s, the number of playable sequences remains the same. While not immediately obvious, this symmetry holds for any number of arguments.

Lemma 3.21. *Let* $\sigma : \{1,\dots,k\} \to \{1,\dots,k\}$ *be a bijection. Then*

$$\chi(x_1,\dots,x_k) = \chi(x_{\sigma(1)},\dots,x_{\sigma(k)}).$$

Proof. It is sufficient to show that $\chi(x_1, \ldots, x_k) = \chi(x_{\sigma(1)}, \ldots, x_{\sigma(k)})$ for any σ with $\sigma(i) = i$ for all but two i's, $i = 1, \ldots, k$. By repeatedly swapping elements, any permutation can be achieved (see also Proposition 2.35 in [5]). We will construct a bijection $f : \text{Pl}(x_1, \ldots, x_k) \to \text{Pl}(x_{\sigma(1)}, \ldots, x_{\sigma(k)})$ for such a map σ. Let $s \in \text{Pl}(x_1, \ldots, x_k)$ be arbitrary. Because $(1, \ldots, k) \subseteq s$, we can divide s into parts $s_1, \ldots, s_k, s_{k+1}$ by choosing s_1 to be the part up to but not including the first 1, s_2 the part from the first 1 up to but not including the first 2, etc. We obtain

$$s = s_1 1 s_2 2 s_3 3 \cdots s_{k-1}(k-1)s_k k s_{k+1}.$$

Note that the s_i may be empty. By construction, s_i does not contain any i's. Now, assuming that for some $i, j \in \{1, \ldots, k\}$ with $i < j$ we have $\sigma(i) = j$ and $\sigma(j) = i$ and that $\sigma(l) = l$ for all $l = 1, \ldots, k$, $l \neq i, j$, we define

$$\begin{aligned} f : & s_1 1 s_2 \cdots (i-1) s_i i \cdots (j-1) s_j j \cdots (k-1) s_k k s_{k+1} \\ & \mapsto s'_1 1 s'_2 \cdots (i-1) s'_j i \cdots (j-1) s'_i j \cdots (k-1) s'_k k s'_{k+1}, \end{aligned}$$

where s'_l equals s_l with all i's replaced by j's and vice versa. In words: we swap s_i and s_j and replace all i's and j's by j's and i's respectively except for the i and j in the subsequence we isolated. First of all, because we do not alter this subsequence, $f(s)$ is $(k, 1)$-playable. Moreover, the amount of l's in $f(s)$ is equal to the amount of l's in s for $l \neq i, j$, i.e., $x_{\sigma(l)} = x_l$ for these l. What remains to be shown for f to be well-defined is that the amount of i's in $f(s)$ is equal to the amount of j's in s and vice versa.

By applying f, we change $x_i - 1$ occurrences of i into occurrences of j and $x_j - 1$ j's are replaced by i's. Exactly one i and one j are left unaltered. As such, the number of i's in $f(s)$ is now $1 + x_j - 1 = x_j = x_{\sigma(i)}$ and the number of j's is $1 + x_i - 1 = x_i = x_{\sigma(j)}$. Therefore, $f(s) \in \text{Seq}(x_{\sigma(1)}, \ldots, x_{\sigma(k)})$, which combined with the fact that $f(s)$ is $(k, 1)$-playable yields that $f(s) \in \text{Pl}(x_{\sigma(1)}, \ldots, x_{\sigma(k)})$ and f is well-defined.

To show that f is a bijection, first note that $(s'_l)' = s_l$ for all l: replacing all i's by j's and vice versa twice results in the sequence we started off with. Similarly, $\sigma(\sigma(i)) = \sigma(j) = i$ and $\sigma(\sigma(j)) = \sigma(i) = j$. We thus find that

$$\begin{aligned} f^2(s) &= f(f(s_1 1 s_2 \cdots (i-1) s_i i \cdots (j-1) s_j j \cdots (k-1) s_k k s_{k+1})) \\ &= f(s'_1 1 s'_2 \cdots (i-1) s'_j i \cdots (j-1) s'_i j \cdots (k-1) s'_k k s'_{k+1}) \\ &= s''_1 1 s''_2 \cdots (i-1) s''_i i \cdots (j-1) s''_j j \cdots (k-1) s''_k k s''_{k+1} \\ &= s_1 1 s_2 \cdots (i-1) s_i i \cdots (j-1) s_j j \cdots (k-1) s_k k s_{k+1} = s, \end{aligned}$$

which shows that $f^2 = f^{-1}$. Hence, f is bijective. □

The following recursive definition of $\chi(x_1, \ldots, x_k)$ will form the basis of the proof for the general case.

Lemma 3.22. *Let $x_1, \ldots, x_k \in \mathbb{N}$ be such that $\sum_{i=1}^{k} x_i = N$. Then*

$$\begin{aligned} \chi(x_1, \ldots, x_k) = \binom{N-1}{x_1 - 1} & \chi(x_2, \ldots, x_k) + \chi(x_1, x_2 - 1, x_3, \ldots, x_k) \\ & + \chi(x_1, x_2, x_3 - 1, \ldots, x_k) + \ldots + \chi(x_1, x_2, x_3, \ldots, x_k - 1). \end{aligned}$$

Proof. Any sequence $s \in \mathrm{Pl}(x_1,\dots,x_k)$ either starts with a 1 or does not start with a 1. If s starts with a 1, we can obtain s by concatenating this 1 and a sequence $t \in \mathrm{Pl}(x_2,\dots,x_k)$ to which we have added $x_1 - 1$ ones. There are $\chi(x_2,\dots,x_k)$ sequences in $\mathrm{Pl}(x_2,\dots,x_k)$ and $x_1 - 1$ ones can be added to such a sequence in $\binom{N-1}{x_1-1}$ ways as we need to choose $x_1 - 1$ spots out of $N - 1$ (all except the first) to place the remaning 1s. This explains the first term. If s does not start with a 1, say s starts with $i \neq 1$, it can be formed by concatenating i and a $(k,1)$-playable sequence containing one i less than s. There are exactly $\chi(x_1,\dots,x_i - 1,\dots,x_k)$ of these sequences, which proves the formula. □

Corollary 3.23. *Let $x_1,\dots,x_k \in \mathbb{N}$ be such that $\sum_{i=1}^k x_i = N$. Then*

$$\psi(x_1,\dots,x_k) = \binom{N-1}{x_1-1}\psi(x_2,\dots,x_k) + \psi(x_1,x_2-1,x_3,\dots,x_k) + \psi(x_1,x_2,x_3-1,\dots,x_k) + \ldots + \psi(x_1,x_2,x_3,\dots,x_k-1).$$

Proof. Note that

$$\binom{N-1}{x_1-1}\binom{N-x_1}{x_2,\dots,x_k} = \frac{(N-1)!(N-x_1)!}{(x_1-1)!(N-x_1)!x_2!\cdots x_k!} = \binom{N-1}{x_1-1,x_2,\dots,x_k}$$

and use Lemma 2.7. □

Lemma 3.21 and Corollary 3.23 allow for an inductive proof of Theorem 3.18, which we will use as guideline for the proof of the general case.

Alternate proof of Theorem 3.18. We will again calculate $\psi(x_1,x_2,x_3)$, now by induction to N. First note that

$$\psi(1,0,0) = T(1,0,0) = \binom{1}{1,0,0} = 1 = 2^0 + \sum_{i=0}^{-1}\binom{1}{i} + \sum_{i=0}^{-1}\binom{1}{i} + \sum_{i=0}^{-1}\binom{1}{i}.$$

By Corollary 3.23, it follows that $\psi(0,1,0) = \psi(0,0,1) = 1$ as well. These are the base cases. Now, assume the theorem holds for all $x_1,x_2,x_3 \in \mathbb{N}_0$ with $x_1+x_2+x_3 = N-1$ and let $X_1,X_2,X_3 \in \mathbb{N}_0$ be such that $X_1+X_2+X_3 = N$. We use Lemma 3.22 for the induction step, where we set $\psi(X_1,X_2,X_3) = 0$ if $X_i < 0$ for any i and apply the result of Theorem 3.17:

$$\begin{aligned}\psi(X_1,X_2,X_3) &= \binom{N-1}{X_1-1}\psi(X_2,X_3) + \psi(X_1,X_2-1,X_3) + \psi(X_1,X_2,X_3-1)\\ &= \binom{N-1}{X_1-1} + 2\cdot 2^{N-1} - 2\sum_{i=0}^{X_1-1}\binom{N-1}{i} - 2\sum_{i=0}^{X_2-2}\binom{N-1}{i}\\ &\quad - 2\sum_{i=0}^{X_3-2}\binom{N-1}{i} - \binom{N-1}{X_2-1} - \binom{N-1}{X_3-1}\\ &= 2^N - 2\sum_{i=0}^{X_1-2}\binom{N-1}{i} - \binom{N-1}{X_1-1} - 2\sum_{i=0}^{X_2-2}\binom{N-1}{i}\\ &\quad - 2\sum_{i=0}^{X_3-2}\binom{N-1}{i} - \binom{N-1}{X_2-1} - \binom{N-1}{X_3-1}.\end{aligned}$$

Now, note that

$$\begin{aligned}
2\sum_{i=0}^{X_j-2}\binom{N-1}{i} &= \sum_{i=0}^{X_j-2}\binom{N-1}{i} + \sum_{i=1}^{X_j-1}\binom{N-1}{i-1}\\
&= \sum_{i=1}^{X_j-2}\binom{N}{i} + \binom{N-1}{0} + \binom{N-1}{X_j-2}\\
&= \sum_{i=0}^{X_j-2}\binom{N}{i} + \binom{N-1}{X_j-2}
\end{aligned}$$

for all $j = 1, 2, 3$, where we use Lemma 2.6 to combine the sums. Using this lemma once more, we find that

$$\binom{N-1}{X_j-1} + \binom{N-1}{X_j-2} = \binom{N}{X_j-1}.$$

Substituting these formulas in the expression we found above, we obtain

$$\begin{aligned}
\psi(X_1, X_2, X_3) = 2^N &- \sum_{i=0}^{X_1-2}\binom{N}{i} - \binom{N}{X_1-1} - \sum_{i=0}^{X_2-2}\binom{N}{i}\\
&- \binom{N}{X_2-1} - \sum_{i=0}^{X_3-2}\binom{N}{i} - \binom{N}{X_3-1}\\
&= 2^N - \sum_{i=0}^{X_1-1}\binom{N}{i} - \sum_{i=0}^{X_2-1}\binom{N}{i} - \sum_{i=0}^{X_3-1}\binom{N}{i}.
\end{aligned}$$

□

In [6], we find a proposal and combinatorial proof for the case $k = 4$, the formula for which we state in Theorem 3.24. However, the inductive proof given for Theorem 3.18 also generalises for the case $k = 4$. We will not work out this proof in detail. Instead, we will discuss the proof of the general formula for arbitrary k in Theorem 3.25 which follows the same structure and view Theorem 3.24 as a special case.

Theorem 3.24. *Let $x_1, \ldots, x_4 \in \mathbb{N}_0$ be such that $\sum_{i=1}^{4} x_i = N$. Then*

$$\begin{aligned}
\chi(x_1, \ldots, x_4) = \binom{N}{x_1, \ldots, x_4} - 3^N + \sum_{i=0}^{x_1-1}\binom{N}{i}2^{N-i} + \ldots + \sum_{i=0}^{x_4-1}\binom{N}{i}2^{N-i}\\
- \sum_{i=0}^{x_1-1}\sum_{j=0}^{x_2-1}\binom{N}{i, j, N-i-j} - \ldots - \sum_{i=0}^{x_3-1}\sum_{j=0}^{x_4-1}\binom{N}{i, j, N-i-j}.
\end{aligned}$$

Before we state the general formula, we introduce some notation. For two sequences $s = (s_i)_{i=1}^{N}$ and $u = (u_i)_{i=1}^{N}$ of the same length, we write $s \preceq u$ if $s_i \leqslant u_i$ for all $i = 1, \ldots, N$. Note that this relation $\preceq$ is in fact a partial order on the set of sequences of length N. We write $|s| = \#\{i \mid s_i > 0\}$, i.e., $|s|$ is the amount of non-negative entries in s. For two numbers $n, m \in \mathbb{N}_0$, we define

$$n \dot{-} m = \begin{cases} n - m, & \text{if } n - m \geqslant 0,\\ 0, & \text{otherwise.}\end{cases}$$

We then write $\Sigma s = \sum_{i=0}^{N}(s_i \dot{-} 1)$ to be the sum of all entries in s lowered by 1. Finally, we define

$$\binom{N}{s} = \binom{N}{s_1 \dot{-} 1, s_2 \dot{-} 1, \ldots, s_N \dot{-} 1, N - \Sigma s}.$$

Note that by definition of Σs, this multinomial coefficient is indeed well-defined. Using this notation, the formula for the general case now reads as follows.

Theorem 3.25. *Let $x_1, \ldots, x_k \in \mathbb{N}$ be such that $\sum_{i=1}^{k} x_i = N$, $k \geqslant 2$. Then*

$$\chi(x_1, \ldots, x_k) = \binom{N}{x_1, \ldots, x_k} - \sum_{\ell=0}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1, \ldots, x_k) \\ |I| = \ell}} \binom{N}{I}(k - \ell - 1)^{N - \Sigma I}.$$

Proof. Let $k \geqslant 2$ be arbitrary. Once more, we find an expression for the amount of unplayable sequences $\psi(x_1, \ldots, x_k)$: we will deduce that

$$\psi(x_1, \ldots, x_k) = \sum_{\ell=0}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1, \ldots, x_k) \\ |I| = \ell}} \binom{N}{I}(k - \ell - 1)^{N - \Sigma I}. \tag{3.4.2}$$

Without loss of generality (cf. Lemma 3.21), we find the base case to be

$$\begin{aligned}
\psi(1, 0, \ldots, 0) &= \sum_{\ell=0}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (1,0,\ldots,0) \\ |I| = \ell}} \binom{N}{I}(k - \ell - 1)^{1 - \Sigma I} \\
&= \binom{N}{(0, 0, \ldots, 0)}(k - 0 - 1)^{1-0} - \binom{N}{(1, 0, \ldots, 0)}(k - 1 - 1)^{1-0} \\
&= k - 1 - (k - 2) = 1.
\end{aligned}$$

For the induction step, we use Corollary 3.23, which gives

$$\begin{aligned}
\psi(x_1, \ldots, x_k) = \binom{N-1}{x_1 - 1} \sum_{\ell=0}^{k-3}(-1)^\ell \sum_{\substack{I \preceq (x_2, \ldots, x_k) \\ |I| = \ell}} \binom{N - x_1}{I}(k - \ell - 2)^{N - x_1 - \Sigma I} \\
+ \sum_{\ell=0}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1, x_2 - 1, \ldots, x_k) \\ |I| = \ell}} \binom{N-1}{I}(k - \ell - 1)^{N - 1 - \Sigma I} + \ldots \\
+ \sum_{\ell=0}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1, x_2, \ldots, x_k - 1) \\ |I| = \ell}} \binom{N-1}{I}(k - \ell - 1)^{N - 1 - \Sigma I}.
\end{aligned} \tag{3.4.3}$$

Now, let $I = (i_1, \ldots, i_k)$ with $I \preceq (x_1, \ldots, x_k)$ and $|I| = \ell$ be arbitrary. We will construct this term in Eq. 3.4.2 using terms from Eq. 3.4.3, showing that every term in Eq. 3.4.2 can be constructed in this way and that every term in Eq. 3.4.3 is used exactly once in this construction.

First, let $i_1 = 0$. For every t with $1 < t \leqslant k$ for which $i_t = 0$, we take the term corresponding to I in the summation which constitutes $\psi(x_1, \ldots, x_t - 1, \ldots, x_k)$, being

$$(-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-1-\Sigma I}. \tag{3.4.4}$$

For every t with $i_t > 1$, we take the term corresponding to $I' = (i_1, \ldots, i_t - 1, \ldots, i_k)$ in the summation of $\psi(x_1, \ldots, x_t - 1, \ldots, x_k)$:

$$(-1)^\ell \binom{N-1}{0, i_2 \dot{-} 1, \ldots, i_t - 2, \ldots, i_k \dot{-} 1, N-1-(\Sigma I - 1)} (k-\ell-1)^{N-1-(\Sigma I-1)}.$$

For every t with $i_t = 1$, we take nothing. Therefore, the contribution for every t with $t > 0$ can be written as

$$\frac{i_t - 1}{N - \Sigma I} (-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-\Sigma I}. \tag{3.4.5}$$

Noting that there are exactly $k - \ell - 1$ values for $t > 1$ for which $i_t = 0$, we obtain as the sum of all terms we took $k - \ell - 1$ times Eq. 3.4.4 plus Eq. 3.4.5 for every t with $i_t > 0$:

$$(-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-\Sigma I} + \frac{\Sigma I}{N - \Sigma I} (-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-\Sigma I}. \tag{3.4.6}$$

Using the definition of $\Sigma I = \sum_{t=1}^{k} (i_t \dot{-} 1) = \sum_{t | i_t > 0} (i_t - 1)$ and Lemma 2.7, we find

$$\binom{N-1}{I} + \frac{\Sigma I}{N - \Sigma I} \binom{N-1}{I} = \binom{N}{I},$$

so that Eq. 3.4.6 turns out to be

$$(-1)^\ell \binom{N}{I} (k-\ell-1)^{N-\Sigma I}, \tag{3.4.7}$$

which is exactly the term corresponding to I in Eq. 3.4.2. Also, note that every term corresponding to an I for which $i_1 = 0$ in a summation of a ψ in Eq. 3.4.3 is used exactly once in this construction.

Now, consider an I for which $i_1 > 0$. Following the same construction, we still deplete all terms in the ψ summations in Eq. 3.4.3, using the same reasoning as before, However, the total of the terms we use in this way now adds up to Eq. 3.4.7 plus an additional term

$$(-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-1-\Sigma I} - \frac{i_1 - 1}{N - \Sigma I} (-1)^\ell \binom{N-1}{I} (k-\ell-1)^{N-\Sigma I},$$

as there is one more t with $1 < t \leqslant k$ for which $i_t = 0$ and we cannot take a term from the non-existent $\psi(x_1 - 1, \ldots, x_k)$ summation. Noting that the second part of this term is 0 for $i_1 = 1$ and following the reasoning behind Eq. 3.4.5, summing the extra terms over all I with $i_1 > 0$ almost telescopes to 0. We are only left with the terms regarding the I with $i_1 = x_1$:

$$\sum_{\ell=1}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1,\dots,x_k) \\ |I|=\ell; i_1=x_1}} (-1)^\ell \binom{N-1}{I}(k-\ell-1)^{N-1-\Sigma I}.$$

In order to correct this, we rewrite

$$\begin{aligned}
&\binom{N-1}{x_1-1}\sum_{\ell=0}^{k-3}(-1)^\ell \sum_{\substack{I \preceq (x_2,\dots,x_k) \\ |I|=\ell}} \binom{N-x_1}{I}(k-\ell-2)^{N-x_1-\Sigma I} \\
&\quad = \sum_{\ell=0}^{k-3}(-1)^\ell \sum_{\substack{I \preceq (x_1,x_2,\dots,x_k) \\ |I|=\ell+1; i_1=x_1}} \binom{N-1}{I}(k-\ell-2)^{N-1-\Sigma I} \\
&\quad = -\sum_{\ell=1}^{k-2}(-1)^\ell \sum_{\substack{I \preceq (x_1,x_2,\dots,x_k) \\ |I|=\ell; i_1=x_1}} \binom{N-1}{I}(k-\ell-1)^{N-1-\Sigma I}.
\end{aligned}$$

We thus find that these terms precisely cancel out the first term in the recursive formula, which completes the proof. □

Note that the structure of the formula for $\chi(x_1,\dots,x_k)$, an alternating sum, suggests that its validity can also be shown using a combinatorial proof based on the principle of in- and exclusion. However, how to construct this proof is as of yet unclear. Furthermore, a comparable example in [7] shows an approach using generating functions, which might be yet another method of proving the statement.

Choosing $x_1 = x_2 = \dots x_k = 1$ and regarding $\psi(x_1,\dots,x_k)$ using Theorem 3.25 yields the following interesting result.

Corollary 3.26. *For arbitrary $k \in \mathbb{N}$:*

$$\sum_{\ell=0}^{k-2}(-1)^\ell \binom{k}{\ell}(k-\ell-1)^{k-\ell} = k! - 1.$$

This expression turns out to be a special case of the formula proven in [8]. However, the main result of Theorem 3.25 is that we can now calculate the amount of $(k,1)$-playable sequences for arbitrary values of the x_i.

Result 3.27. Theorem 3.25 allows us to compute the amount of playable sequences if we follow the distribution of the numbers in classic Hanabi as in Example 2.10. We find $\chi(3,2,2,2,1) = 5934$. With the total number of admissible sequences being $T(3,2,2,2,1) = 75600$, only $\frac{5934}{75600} \approx 0.078$ of all possible stacks containing the specified numbers of cards are $(5,1)$-playable.

3.5 Dynamic programming

While Theorem 3.25 provides a complete characterisation of the $(k, 1)$-playable sequences, it does not cover games where the hand size is larger than 1. Indeed, reasoning about (k, m)-playability for $m > 2$ turns out to be much harder than the case $m = 1$ as Theorem 3.13.2 no longer holds. Therefore, instead of taking a theoretic combinatorial approach, we will turn to the use of the computer to determine the playability of a given sequence. This process is based on the observation that we can determine whether a sequence is (k, m)-playable by searching for and playing the first 1 in the sequence (discarding cards on the way if necessary), subtracting 1 from the rest of the elements in the sequence and then wondering whether the remaining sequence is $(k-1, m)$-playable. This statement is captured in the following theorem, where we use the notation as introduced in Section 2.1.

Theorem 3.28. *Let $S = (s_i)_{i=1}^N$, $s_i \in \mathbb{Z}$ be an ordered sequence. Assume $1 \in S$ and write $S = t1u$, where t is the part of S up to the first 1. S is (k, m)-playable if and only if $S' = (t' - 1)(u - 1)$ is $(k - 1, m)$-playable for some subsequence $t' \subseteq t$ of length not greater than $m - 1$.*

Proof. Assume S is (k, m)-playable. Let 1_f be the first occurrence of 1 in S, let s_f be the position of 1_f and write $S = t1_f u$. We distinguish between two cases.

First, assume that $f \leqslant m$, i.e., that the first 1 is contained in the starting hand of the player, i.e., $1_f \in P_1$ after initialisation. We can then play 1_f to $F[1]$ as action in our first turn. After doing so, we then have $P = \{(s_i)_{i=1}^h \backslash 1_f\}$. Note that this would exactly be the starting hand of the player if the game had begun with the sequence $S' = tu$. Naturally, wondering whether we can now play a sequence $(2, \ldots, k)$ is the same as wondering whether we could play a sequence $(1, \ldots, k-1)$ if using the stack $(t-1)(u-1)$. Moreover, the length of t is at most $m - 1$ by assumption.

Now, assume $f > m$. It is clear that the first time that a card can be played is immediately after 1_f has entered the player's hand P_1. At this point, we have $P = \{t'1_f\}$ with t' being a subsequence of t of length $m-1$. This would exactly be the starting hand of the player if the algorithm had started with the stack $S' = t'1_f u$. Because the initial stack S is (k, m)-playable by assumption, there must exist a t' for which S' must be (k, m)-playable. Then $(t' - 1)(u - 1)$ must be $(k - 1, m)$-playable as well.

Finally, suppose $(t' - 1)(u - 1)$ is $(k - 1, m)$-playable for some subsequence t' following the given requirements. Let $S = t'1u$. Because the length of t' is at most $m-1$, the added 1 will be in the starting hand of the player. On the first turn, it can be played to $F[1]$, after which P will be the same as the original opening hand, but with the values of all cards increased by 1. As we could play a sequence $(1, \ldots, k-1)$ using the original stack, we can thus play a sequence $(2, \ldots, k)$ from now. With the 1 already having been played, this proves the (k, m)-playability of S. □

Note that elements of the sequence may be reduced to values of 0 or lower. In practice, this essentially means that the respective cards cannot be played.

However, they cannot be completely removed from the stack as they might act as a buffer which prevents the stack from emptying.

Theorem 3.28 allows us to recursively determine whether a sequence S is (k, m)-playable: we search for the first 1 in S, remove this 1 and perhaps some other numbers from S, lower all elements in S by 1 and repeat the process. This recursive definition gives rise to an implementation using dynamic programming, the theory of which is explained in Chapter 8 of [9]. The result is the following algorithm, employing properties from Theorem 3.13 as well.

Algorithm 3.29 (Determine the (k, m)-playability of a single-colour sequence)**.** Given an ordered sequence $S = (s_i)_{i=1}^N$, $s_i \in \mathbb{Z}$ and integers $k, m \in \mathbb{Z}_{\geqslant 1}$, this algorithm decides whether S is (k, m)-playable.

1. Initialisation. Set $n := N$, $p := k$. If $n - m + 1 < k$, return NO.
2. (a) Let i be the position of the first 1 in S or set $i := \infty$ if $1 \notin S$. If $i = \infty$, return NO. If $p = 1$, return YES. If $n - i < p - 1$, return NO.
   (b) If $i \leqslant m$, set $S := (S \backslash s_i) - 1$ and $n := n - 1$ and $p := p - 1$. Go to step 2.
   (c) Let $S = t1u$, where t is of length $i - 1$. For all subsequences $t' \subseteq t$ of length $m - 1$, recursively invoke step 2 with $S := t'u - 1$, $n := n - i + m - 1$ and $p := p - 1$. If one of the subsequences t' results in YES, return YES. Otherwise, return NO.

Example 3.30. We will determine whether $S = (4, 2, 3, 1, 3, 2, 1, 1)$ is $(4, 2)$-playable using Algorithm 3.29. In the initialisation, we have $n = 8$ and $p = 4$. As $8 - 2 + 1 = 7 \nless 4$, we continue to start iterations of step 2.

1. We find $i = 4$ and return nothing in step a. Skipping step b as $4 \nleqslant 2$, we execute step c. We recursively test all subsequences $t' \subseteq (4, 2, 3)$ of length 1. In any case, set $n = 5$ and $p = 3$.
   (a) Take $S = (4, 3, 2, 1, 1) - 1 = (3, 2, 1, 0, 0)$. We find $i = 3$. As $2 \nless 2$, we continue to step b. Here, $3 \nleqslant 2$, so we try all subsequences $t'' \subseteq (3, 2)$ of length 1, always setting $n = 3$ and $p = 2$.
      i. Take $S = (3, 0, 0) - 1 = (2, -1, -1)$. We now find $i = \infty$ and return NO.
      ii. Take $S = (2, 0, 0) - 1 = (1, -1, -1)$. We again find $i = \infty$ and return NO.
   (b) Take $S = (2, 3, 2, 1, 1) - 1 = (1, 2, 1, 0, 0)$.
      i. We find $i = 1$. Now we see $4 \nless 3$, so we continue to step b. Here, $1 \leqslant 2$, so we set $S = (2, 1, 0, 0) - 1 = (1, 0, -1, -1)$, $n = 4$ and $p = 2$ and return to step 2.
      ii. We find $i = 1$. As $3 \nless 1$, we continue to step b, where we again find $1 \leqslant 2$. So $S = (0, -1, -1) - 1 = (-1, -2, -2)$, $n = 3$ and $p = 1$.
      iii. We obtain $i = \infty$ and return NO.

(c) Take $S = (3, 3, 2, 1, 1) - 1 = (2, 2, 1, 0, 0)$. We find $i = 3$ and $2 \nless 2$, so continue to step b. Finding $3 \nleq 2$, we skip this step and to go step c, where we try all subsequences $t' \subseteq (3, 3)$ of length 1, setting $n = 3$ and $p = 2$. Note that (3) is the only subsequence available: taking $S = (3, 0, 0) - 1 = (2, -1, -1)$, we find $i = \infty$ in step a and return NO.

We thus find that S is not $(4, 2)$-playable.

The fact that we need to test all possible subsets $t' \subseteq t$ to see whether a stack is playable illustrates the fact we already discovered in Section 3.2: even though we have perfect information in open games, it is still hard to find a strategy which plays a playable game. Even so, calculations using the algorithm give us the following result.

Result 3.31. Using tables to keep track of the playability of smaller sequences, one can compute the amount of (k, m)-playable sequences of length 10 for given amounts of 1s, 2s, etc. The results can be found in the tables in appendix A.

Note that the largest fraction of playable stacks is encountered when choosing three 1s, two 2s, two 3s, two 4s and one 5, which is exactly the configuration used in classic Hanabi. For a hand size of 5, approximately 0.8778 of all initial stacks are playable when using these amounts of cards. Note that this fraction lies below 1 even though there is now enough space to store all possible cards 1 through 5 and prevent a crucial card from being discarded. The unplayable stacks must thus be unplayable only because the stack runs out before the filling of the field stacks was completed.

Furthermore, note that while the symmetry proven in Lemma 3.21 is indeed apparent for hand size 1, this symmetry does not seem to hold for larger hand sizes. Indeed, we find that the portion of playable stacks when taking six 5s and one card of every other value is approximately 0.2571, while picking six 1s and one copy of every other number results in a fraction of 0.4764. It thus seems that the addition of the possibility of storing cards in hand disrupts the exchangeablilty of the values.

3.6 Playability of multi-colour sequences

Definition 3.10 of the playability of a single-colour sequence is easily extended to cover multi-colour sequences, i.e., stacks consisting of cards of several different colours. Note that we again take our values in $\mathbb{Z}$ instead of $\mathbb{N}$ to allow for non-positive values.

Definition 3.32. Let $S = (s_i)_{i=1}^N$, $s_i = (x_i, y_i) \in \mathbb{Z} \times \mathbb{Z}$ be a sequence. Let $n = \max_{i=1}^N x_i$ and $c = \max_{i=1}^N y_i$. S is (k, m)-*playable* if there exist an open game $L = (n, c, 1, m, S, 0, 0)$ and open strategy x_o such that $v(x_o, L) = k$.

Note that Properties 3.13.1 and 3.13.4 still hold for multi-colour sequences, with identical proof. An analogon of 3.13.3 can be constructed if we note that in a game with a single player and hand size 1, building the stacks in the different colours occurs in a completely parallel way, i.e., the colours do not interfere. Therefore, if the stack is sorted by colour without altering the numerical order of the cards within a single colour, playability is preserved.

Theorem 3.33. *Let $S = (s_i)_{i=1}^N$, $s_i = (x_i, y_i) \in \mathbb{Z} \times \mathbb{Z}$ be a sequence. Let S' be the sequence S sorted by non-descending y_i using a stable sorting method. Then S is $(k,1)$-playable if and only if S' is $(k,1)$-playable.*

Proof. First, suppose S is $(k,1)$-playable. Letting P_t, $t \geqslant 1$ be the hand of the player at the start of turn t, we have $P_t = \{s_t\}$ as in the proof of Theorem 3.13.1. Because S is $(k,1)$-playable, we must thus be able to find subsequences $((1,y),(2,y),\ldots,(k,y)) \subseteq S$ for all $y = 1,\ldots,c$. Because the sorting method used is stable with respect to the values of the cards, these subsequences are conserved when building S'. Therefore, S' is playable. The proof in the other direction is similar. □

Corollary 3.34. *Let $S = (s_i)_{i=1}^N$, $s_i = (x_i, y_i) \in \mathbb{Z} \times \mathbb{Z}$ be a sequence with $c = \max_i y_i$ and let $Y_j = (x \mid (x,j) \in S) \subseteq S$, $1 \leqslant j \leqslant c$ be subsequences. S is $(k,1)$-playable if and only if there exist $k_j \in \mathbb{N}_0$, $1 \leqslant j \leqslant c$ with $\sum_{j=1}^c k_j = k$ such that Y_j is $(k_j,1)$-playable for all j.*

This statement gives a simple characterisation of $(k,1)$-playability of multi-colour stacks. For (k,m)-playability with $m > 1$ we can determine whether a stack is playable by employing a recurrence relation like in the single-colour case, which is captured in the following analogon of Theorem 3.28.

Theorem 3.35. *Let $S = (s_i)_{i=1}^N$, $s_i = (x_i, y_i) \in \mathbb{Z} \times \mathbb{Z}$ be a ordered sequence. Assume $(1,y) \in S$ for a certain $y \in \mathbb{N}_0$. Let $1_S = \min\{i \mid (x_i,y_i) \in S\}$ be the position of the first 1 in S and write $S = t(x_{1_S}, y_{1_S})u$. S is (k,m)-playable if and only if $S' = (t' -_{y_{1_S}} 1)(u -_{y_{1_S}} 1)$ is $(k-1,m)$-playable for some subsequence $t' \subseteq t$ of length $m-1$, where $s_i -_y 1 = (x_i - \mathbb{1}_{\{y_i = y\}}, y_i)$.*

Proof. Similar to the proof of Theorem 3.28. □

Again, this recursive definition of (k,m)-playability of a sequence gives rise to an algorithm. Now, we search for the first 1 in S, remove this 1 and perhaps some other numbers from S, lower all elements of the same colour as the removed 1 in S by 1 and repeat the process.

Algorithm 3.36 (Determine the (k,m)-playability of a multi-colour sequence)**.** Given a sequence $S = (s_i)_{i=1}^N$, $s_i = (x_i, y_i) \in \mathbb{Z} \times \mathbb{Z}$ and integers $k, m \in \mathbb{Z}_{\geqslant 1}$, this algorithm decides whether S is (k,m)-playable.

1. Initialisation. Set $n := N$, $p := k$. If $n - m + 1 < k$, return NO.
2. (a) Let i be the position of the first 1 in S or set $i := \infty$ if $1 \notin S$. If $i = \infty$, return NO. If $p = 1$, return YES. If $n - i < p - 1$, return NO.
   (b) If $i \leqslant m$, set $S := (S \backslash s_i) -_{y_i} 1$ and $n := n - 1$ and $p := p - 1$. Go to step 2.
   (c) Let $S = t1u$, where t is of length $i - 1$. For all subsequences $t' \subseteq t$ of length $m - 1$, do $S := t'u -_{y_i} 1$, $n := n - i + m - 1$, $p := p - 1$, go to step 2.

Example 3.37. Using Algorithm 3.36, we will determine whether

$$S = ((2,1),(1,1),(3,2),(1,2),(3,1),(2,2),(1,1))$$

is $(4,2)$-playable. In the initialisation, we have $n = 7$ and $p = 4$. As $7 - 2 + 1 = 6 \nless 4$, we continue to start iterations of step 2.

1. We find $i = 2$. As $7 - 2 = 5 \nless 4 - 1 = 3$, continue. In step b, we find $i = 2 \leqslant 2$, so $S := ((1,1),(3,2),(1,2),(2,1),(2,2),(0,1))$, $n := 6$, $p := 3$ and go back to 2a.
2. Now $i = 1$. $6-2 = 4 \nless 3-1 = 2$, continue. In step b, we find $i = 1 \leqslant 2$, so $S := ((3,2),(1,2),(1,1),(2,2),(-1,1))$, $n := 5$, $p := 2$ and return to step 2a.
3. $i = 2$. Because $5-2 = 3 \nless 2-1 = 1$, continue. In step b, we find $i = 2 \leqslant 2$, so $S := ((2,2),(1,1),(1,2),(-1,1))$, $n := 4$, $p := 1$ and return to step 2a.
4. $i = 1$. As $p = 1$, we return YES.

We thus find that S is $(4,2)$-playable.

In practice, it quickly becomes infeasible to calculate the amount of (k,m)-playable sequences for stacks consisting of given cards, as the amount of permutations rises superexponentially. For example, the total amount of stacks of size 12 using at most three numbers and two colours is $(2 \cdot 3)^{12} \approx 2$ billion. Even if the amount of the specific cards is fixed beforehand, we are still left with many configurations: taking two copies of every card in the given example, we still find $\frac{12!}{(2!)^6} \approx 7$ million possbilities. Therefore, the suggested approach of dynamic programming is no longer effective, as the tables in which the data on the sequences is stored grow too large to fit into memory. The fraction of playable sequences might still be approximated using simulations, however, which might form an area for future research.

3.7 Multi-player Hanabi

When dealing with more than 1 player, the option of passing becomes relevant.

Example 3.38. Let $L_1 = (4,1,2,2,S_0,0,0)$ and $L_2 = (4,1,2,2,S_0,\infty,0)$ be two open games of Hanabi with $S_0 = (4,3,2,1,1,1,1,1)$. We have $P_1 = \{3,4\}$ and $P_2 = \{1,2\}$. In L_1, on the first turn of player 1, as he may not pass, he must play or discard a card. This card being a 3 or 4, it is evident that the maximum score of 4 cannot be achieved in L_1, i.e., L_1 is not playable.

In L_2, player 1 may spend his first turn doing nothing. Player 2 is then given the opportunity to play the 1 from his hand. Continuing in this fashion, passing where necessary, it is obvious that a stack of 1 up to 4 can be formed, hence L_2 is playable.

For hand size 1 and allowing no passes, a multi-player game can be reduced to a single-player game.

Theorem 3.39. *Let $L = (n, k, p, 1, S_0, 0, f_0)$ be an open game of Hanabi. Let $L' = (n, k, 1, 1, S_0, 0, f_0)$. Then L is playable if and only if L' is playable.*

Proof. Assume L is playable and let x_o be the strategy that plays L. Note that on turn i, card s_i is played or discarded by some player, who then draws card s_{i+p} from the stack if $i + p < N$ (otherwise the hand remains empty). We can thus define $x'_o = x_o$ as a strategy for the single player which plays L'. The proof in the other direction is similar. □

With larger hand sizes or non-trivial amounts of available passes, the analysis of the playability of multi-player games becomes excessively difficult. For these choices $h > 1$ and $t \in (0, \infty]$, there are many possible strategies and there does not seem to be a simple method of deciding whether one of these strategies succesfully plays the game. In fact, this problem seems equivalent to that of determining an optimal strategy. We decide to now focus on that problem instead, leaving the further calculations on playability for possible future research.

4 Strategies for classic Hanabi

In this section, we will turn to answering the second main question: given a random configuration of the game of Hanabi, what is a good strategy to play? After having defined two different optimality criteria, we will outline several strategies. We will then test these strategies on the classic game of Hanabi as defined in Example 2.10 and discuss the results of these experiments.

4.1 Approach

We now return to "regular" closed games of Hanabi as given in Definition 2.9 and pose the question: given a game of Hanabi $H = (n, k, p, h, S_0, t_0, f_0)$ with a randomly permuted stack S_0, what strategy x is optimal to play? Before being able to start to work on an answer, we must first make sure that the question is well-defined. Indeed, what is an optimal strategy? One obvious definition would be to define an optimal strategy x^* by the strategy which achieves the best average score, i.e.,

$$x^* = \text{argmax}_x \, \mathbb{E}\left(\overline{v}\left(x, H\right)\right),$$

where the expectation is taken over all possible permutations of the stack S_0 in H. However, a bolder player may not be interested in achieving a high score on average, but to optimise his chances of achieving a maximum score, perhaps risking a higher probability of failure. To properly express this goal, we introduce the following notation.

Definition 4.1. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi and x a strategy. Then $\mathbb{P}(x, H)$ is the probability that a score of $n \cdot k$ is reached in H by playing x.

Using this definition, an optimal strategy x^* can then also be defined as a strategy which maximises the probability of achieving a perfect score:

$$x^* = \text{argmax}_x \, \mathbb{E}\left(\mathbb{P}\left(x, H\right)\right).$$

Again, the expectation is taken over all possible permutations of the stack S_0 in H. We will see that a strategy which is optimal for the one optimality criterion might not be optimal for the other and vice versa.

Two well-known methods that could be tried to search for optimal strategies are the Monte-Carlo Tree Search, for which the basis is explained in [10] and several extensions are proposed in [11], and the Minimax algorithm with alpha-beta pruning as described in [12]. However, both these methods face some problems. First of all, one of the most prominent features of Hanabi is that of grossly imperfect information. Indeed, players have very limited knowledge of the cards in their hand. Moreover, hints that are given carry more information than visible at first sight. If three 1s have already been played, for example, and John tells Mary that she has exactly one 2 in her hand, she may well assume that this 2 will fit one of the three 1s without having been told so. Furthermore, a truly random player will end the game with zero score by making too many mistakes very quickly. These three reasons make the implementation of the Monte-Carlo method a difficult task. Finally, disregarding the storage of hint information completely, the different configurations of the cards already give a magnitude of 10^{50} different states of the game and every player has between 20 and 50 actions per turn, depending on the number of players. This huge size of the search space and large branching factor make algorithms like Minimax infeasible: there are simply too many options to consider.

Therefore, we will implement a different method in our search for good strategies. We will use our experience in playing the game in daily life to design, implement and compare several different strategies consisting simple guidelines on how to play the game. These strategies might for example outline rules for when to play or discard a card or for the hints to be given. These rules might involve probabilities, e.g., if I am 80% certain that I can play a card, should I always do so? Or should I try to play the card in only a percentage of the situations?

Before we introduce these rules and probabilities in the next section, it is useful to have the following three definitions regarding cards.

Definition 4.2. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi with current field stacks F and let $c = (x, y) \in \text{Cards}(H)$ be a card. Then c is *playable* if $\#F[y] = x - 1$ or $F[y] = \varnothing \wedge x = 1$ holds.

Note that the definition of playability of a card is different from and in fact has nothing to do with the definition of playability of a game.

Definition 4.3. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi with current stack S, hands P_j and field stacks F and let $c = (x, y) \in \text{Cards}(H)$ be a card. Then c is *worthless* if $\#F[y] \geqslant x$ or $\#F[y] < x \wedge (\exists z : \#F[y] < z < x \wedge (z, y) \notin S \cup \bigcup_{j=1}^{p} P_j)$ holds.

Definition 4.4. Let $H = (n, k, p, h, S_0, t_0, f_0)$ be a game of Hanabi with current stack S and hands P_j and let $c = (x, y) \in \text{Cards}(H)$ be a card. Then c is *unique* if $\#\{(x, y) \in S \cup \bigcup_{j=1}^{p} P_j\} = 1$.

Example 4.5. Let H be classic Hanabi as in Example 2.10 and let the field stack $F[1]$ be given by $F[1] = (1, 2, 3)$. At this point, all cards of the form $(x, 1)$ with $x \leqslant 3$ are useless, as the stack has already been filled to a value of 3 and no second stack of the same colour may exist. The cards $(4, 1)$ are playable, as these might be added to $F[1]$ when played. If both these cards are already in the discard pile, we say that $(5, 1)$ is useless as well, as it can never be played because the 4 which needs to be played first misses. Note furthermore that $(5, 1)$ is always unique by the definition of classic Hanabi.

Intuitively, it seems a good idea to play any playable cards to advance the field stacks, discard any worthless cards to free up hints and keep non-worthless unique cards to prevent stacks from being blocked. Hints should thus give information about these three types of cards. We will make this more precise in the next section.

4.2 Parameters

In Definition 2.17, a decision rule was defined as a probability distribution on the possible actions based on the currently available information. We will construct a stationary strategy using several rules of thumb and probabilities, using the following algorithm as a basis.

Algorithm 4.6 (Playing a game of Hanabi using rules of thumb)**.** Given a game $H = (n, k, p, h, S_0, t_0, f_0)$ of Hanabi with $S_0 = (s_i)_{i=1}^N$, this algorithm describes the playing of the game using some preset rules. The details of every action can be found in Algorithm 2.16.

1. Initialisation.
2. Every turn, do:
   (a) If the active player's hand contains at least one playable card, play one of these cards chosen randomly.
   (b) Else, if the active player's hand contains at least one worthless card, discard one of these cards chosen randomly.
   (c) Else, if there are any hints left, give a hint.
   (d) Else, discard a card.

The exact execution of the steps in the algoritm depends on the chosen rules and parameters. We will consider stationary strategies using decision rules of the form $\omega = (\omega_p, \omega_s, \omega_d, \omega_h, \rho_h, \rho_d)$, for which we will now intoduce the parameters step by step.

Parameters

1. Using all knowledge available, a player may calculate the probability that a card in his hand is playable. If this probability exceeds a predetermined threshold $\omega_p \in [0, 1]$, a player will mark this card as playable in step 2a.
2. If $\omega_p < 1$, we might make errors when attempting to play a card. The parameter $\omega_s \in \{0, 1\}$ controls whether we want to play safe or take the risk of making three errors: if $\omega_s = 1$ and two errors have been made, we will only play a card in step 2a if the active player is certain of its playability.
3. The parameter $\omega_d \in [0, 1]$ mimics the role of ω_p, now in step 2b: if the active player deduces that the probability of a certain card being worthless is at least ω_d, it is marked as such.
4. In the given base algorithm, a hint is always given in step 2c if possible. However, it might be better to sometimes discard even though a hint is available. To explore this possibility, we introduce the parameter $\omega_h \in [0, 1]$: if a hint is available, it is given with probability ω_h. Otherwise, we turn to step 2d immediately.
5. In step 2c, the active player gives a hint to another player. The way in which the given hint is chosen is defined by the hint rule ρ_h, for which several choices are outlined below.
6. Finally, in step 2d, the active player discards a card. The way in which the card to be discarded is selected is determined by the discard rule ρ_d.

Here, the hint and discard rules ρ_h and ρ_d can be picked from the following alternatives, respectively.

Hint rules ρ_h

1. Random. A hint about a random value or colour is given to a random fellow player.
2. Most voluminous. A hint is given which provides information about most cards at once, i.e., a hint providing information about three cards is preferred over a hint providing information about two.
3. First playable card, then most voluminous. Looking in turn order, a hint is given on the first playable card found. If not yet given, a hint on the value is preferred over a hint on the suit. If a player holds multiple playable cards, a hint on a lower-valued card is preferred. If no playable cards are in view, the most voluminous rule is used.
4. First playable card, then first worthless card, then most voluminous. An attempt is made to give a hint on the first playable card as in 3. If no playable cards are available, a hint is given on the first worthless card in sight following the same rules. If also no worthless card can be seen, the most voluminous rule is used.

Discard rules ρ_d

1. Random. A random card is discarded, disregarding all knowledge available.
2. Most worthless. For every card in hand, the probability of it being worthless is calculated using the available information. The card with the highest probability of being worthless is discarded.
3. Oldest. The card which has been present in the active player's hand for the most turns is discarded.
4. Least unique. For every card in hand, the probability of it being unique is calculated using the available information. The card with the lowest probability of being unique is discarded.

Of course, many other hint and discard rules could be thought of. In the experiments to follow, however, we will choose ρ_h and ρ_d from the four hint rules and four discard rules described above and compare the resulting decision rules for several different choices of the parameters ω_i. Here, by slight abuse of notation, we will write $\rho_h = 1$ to describe a strategy using the first hint rule described above, for example.

4.3 Experiments and results

We take classic Hanabi as in Example 2.10 with $p = 3$ and thus $h = 5$. For a decision rule ω, we take the stationary strategy $x = (\omega)_t$. To calculate the average value $\overline{v}(x, H)$ of a strategy x, we play the same game H ten times in a row and compute the average score. To then approximate the value $\mathbb{E}(\overline{v}(x, H))$, we play 10000 different games in this fashion and average the amount of points

Discard rule ρ_d	Hint rule ρ_h			
	1	2	3	4
1	$(5.50562, 14, 0)$	$(12.5932, 22, 2)$	$(13.1048, 21, 4)$	$(10.7417, 20, 3)$
2	$(5.77127, 16, 0)$	$(12.6614, 22, 1)$	$(12.8471, 21, 3)$	$(10.7758, 21, 2)$
3	$(4.96048, 14, 0)$	$(12.2293, 20, 5)$	$(12.6672, 21, 5)$	$(10.2653, 20, 3)$
4	$(4.86980, 14, 0)$	$(12.0331, 20, 3)$	$(12.3029, 21, 4)$	$(10.3384, 20, 3)$

Figure 1: Scores obtained for $\omega_p = \omega_s = \omega_d = \omega_h = 1$.

obtained. With these values, the simulations testing a strategy take approximately one minute to complete. In addition to calculating the average score, we also calculate the minimum and maximum score obtained for every tried decision rule ω. The full results can be found in the tables in Appendix B.

As it takes slightly over a minute to calculate the scores for a decision rule, not all rules could be tested. Therefore, in our search for an optimal rule, we iteratively chose which values to try. Intuitively, the hint and discard rules should have the most impact on the score obtained. Therefore, we started off trying all 16 combinations of these rules while keeping the other parameters constant at value 1. In the table in Figure 1, the resulting tuples (average, maximum, minimum) score are shown.

It is very clear that hint rules 2 and 3 are superior to rules 1 and 4. Between these two rules, the rule of giving a hint about the next playable card seems to be optimal, but only slightly. The result is to be expected: by giving information on playable cards, fellow players obtain knowledge on which cards to try and add to the field stacks, which progresses the score.

The differences between the various discard rules are smaller. According to these experiments, rules 1 and 2 seem to slightly outperform rules 3 and 4, which is confirmed by the values in the appendix. It is not surprising that rule 2 performs well: discarding worthless cards does not damage the maximum possible obtainable score. It is slightly surprising that rule 1 performs better than rules 3 and 4, though. Indeed, it seems to be better to discard randomly than to keep track of the oldest card or to try and avoid unique cards.

As can be found in the appendix, the combination of the hint and discard rules $\rho_h = 3$ and $\rho_d = 2$ provides room for the highest average score. The values obtained using these rules are shown in Figure 2, where $\omega_d = \omega_h = 1$ are kept constant and ω_p and ω_s are varied.

We see that taking some risks in playing cards pays off. To be exact, playing a card when at least 60% sure of its playability results in the best average score and playing when at least 50% sure gives the best maximum score. This trend is also apparent for other combinations of hint and discard rules. The fact that the highest average scores are obtained at a lower risk value than the highest maximum scores can be explained by noting that by taking a higher risk in playing cards, more cards can potentially be played faster, but also more unique cards might be turned to waste.

Furthermore, it is apparent that while taking some risk is recommended, taking

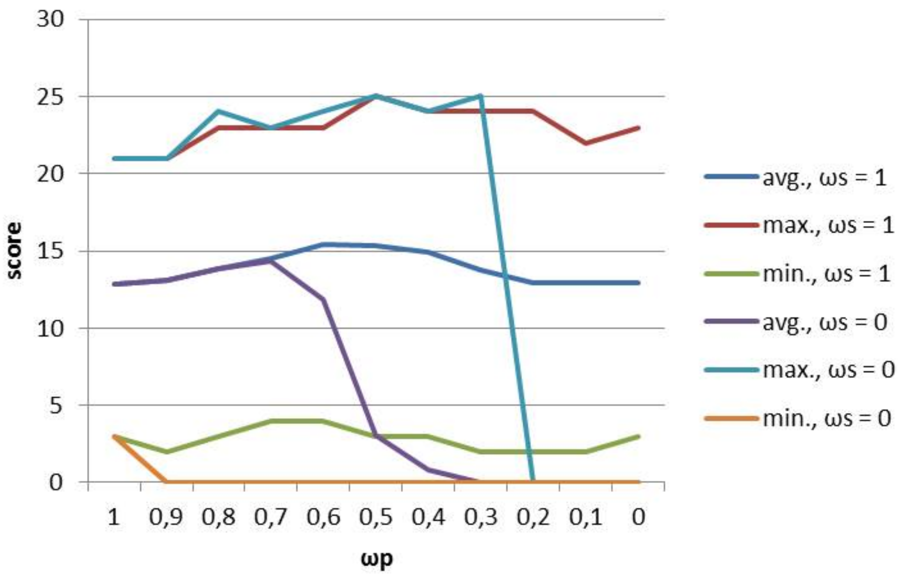

Figure 2: Scores obtained for $\rho_d = 2$, $\rho_h = 3$, $\omega_d = \omega_h = 1$.

too much risk is suboptimal if in search of the best average score. If taking the risk of making a third error and thus ending the game with a score of 0 points, the average score obtained is larger only for the highest values of ω_p. If more uncertainty is accepted and ω_p is lowered, the score obtained decreases rapidly. For $\omega_p \leqslant 0.2$, the choice $\omega_s = 0$ even results in an average score of 0 as every game is ended by making a third error, because the players are too reckless when playing cards. Do note that the maximum scores achieved are generally higher for $\omega_s = 0$: taking the risk of ending the game with a score of zero does bring a higher probability of obtaining the maximum score, as well.

Now, having established that the choice $\rho_h = 3$ and $\rho_d = 2$ is promising and that values of $0.5 \leqslant \omega_p \leqslant 0.7$ and $\omega_s = 1$ provide the best average score among the values we tested, we keep these rules and parameters fixed while varying ω_d. Some results can be found in Figure 3.

We see that for values $0.8 \leqslant \omega_d \leqslant 1.0$, the scores obtained do not vary much. In fact, it seems that all fluctuations should be contributed to the randomness of the games being sampled — there is no clear trend. For lower values of ω_d, the scores obtained steadily decrease. It thus seems best to only discard cards willingly when completely sure of their worthlessness. This result is further established by the experiments where ω_h is varied: keeping all other parameters fixed, we obtain Figure 4.

As was to be expected from the previous experiments, Figure 4 also shows that it does not seem to be profitable to discard a card using the discard rule if there is a hint available. A player should give this hint instead.

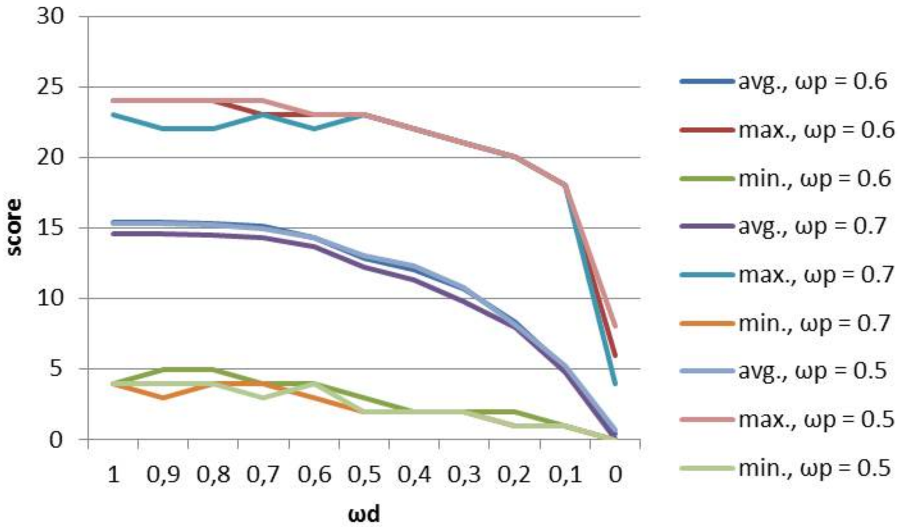

Figure 3: Scores obtained for $\rho_d = 2$, $\rho_h = 3$, $\omega_s = \omega_h = 1$.

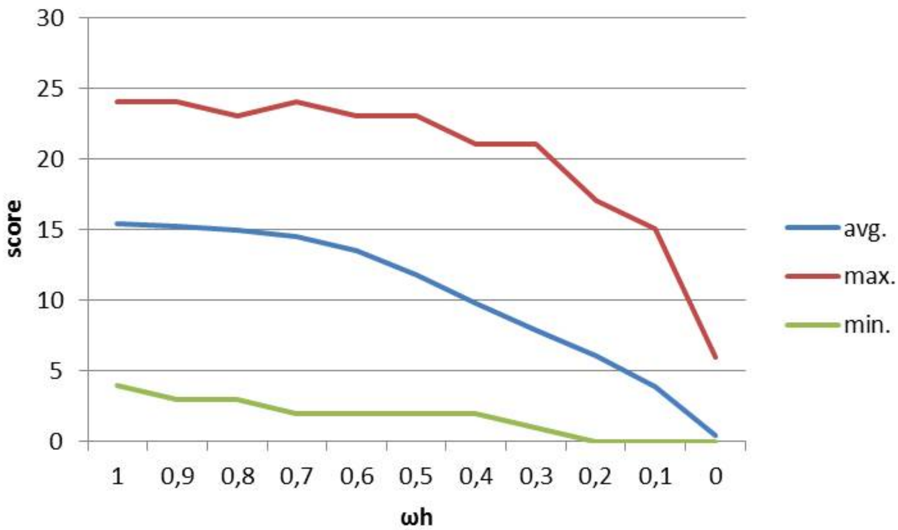

Figure 4: Scores obtained for $\rho_d = 2$, $\rho_h = 3$, $\omega_p = 0.6$, $\omega_s = \omega_h = 1$.

4.4 Discussion

As discussed previously, hints in Hanabi often convey more information than explicitly clear. While the option of playing a card when not 100% sure about its playability was implemented in the parameter ω_p, this implicit information in the hints was not taken into account. It might be interesting to see whether considering this implicit information makes any difference. We could for example record whether a hint was given about multiple cards or one card in particular and introduce a hint rule to give information about single playable cards when possible.

Another property of the real game which was not captured in these decision rules is the fact that players anticipate on each other's actions. For example, in practice it seems smart to remember which card in a player's hand is the oldest card and to agree upon always discarding the oldest card if a semi-random discard action is necessary. In this way, players can predict the actions of each other and prevent the discarding of a unique card only when necessary, which makes deciding what to do easier. It could be interesting to implement this as well.

Furthermore, as mentioned before, the size of the search space makes it impossible to test each and every value of all parameters. Here, the choice of which values to test was made empirically and based on the results of previous experiments. Instead, one could consider using more structured methods of exploring the search space. We might for example think of natural computing methods like particle swarm optimisation or simulated annealing — explained in [13] and [14] respectively — to search for optimal solutions.

Moreover, the heuristics which were implemented turn out to result in a maximum score of 25 points only very seldomly, which makes the analysis of $\mathbb{P}(x, H)$ difficult if not outright impossible. The only conclusion on this subject which can be drawn so far is that strategies involving more risk give a higher probability of obtaining a maximum score while compromising the average score. Though the general lack of maximum scoring might be attributed to a sizeable portion of the sampled games not being playable cf. the previous section, it is likely that other and better strategies may more often result in a maximum score. If so, this could lead to a more detailed and interesting analysis of the optimality criterion $\mathbb{P}(x, H)$.

Finally, one could experiment some more with the base parameters of the game. Games with other amounts of players, hints and errors may result in different scores and other optimal strategies, which could also be an interesting field for further research.

5 Conclusions and further research

In this thesis, we consider the game of Hanabi with two main questions in mind. The first of these questions is which fraction of the initial configurations of a game of Hanabi are playable, i.e., for which percentage of the permutations of the initial stack can a perfect score be obtained? In Section 3, we start out by proving that for the question of playability, we can consider open games instead of closed games without loss of generality. We then continue by looking at games of increasing complexity, starting off with single-colour games played by a single player with hand size 1. For this situation, the question of playability gives rise to a combinatorial problem regarding sequences, for which the main result is stated in Theorem 3.25.

Subsequently, we increase the complexity of the games step by step, at first increasing the hand size of our single player to a value greater than 1. For this type of game, we calculate the amount of playable sequences for various compositions of the initial stack using a dynamic programming algorithm based on a recurrence relation. We discover that for multi-colour games, a recurrence relation still holds, but the implementation of the ensuing algorithm turned out to be problematic because of memory issues. Finally, we show that for the simplest choices for the hand size and amount of passes, multi-player games are equivalent to single-player games.

In Section 4 we turn to the second main question: given a random initial configuration of the game, what strategy is good or even optimal? We introduce two optimality criteria, the former optimising the average score obtained by employing a specific strategy and the latter valuing the probability of obtaining a perfect score. Subsequently, we look at a set of heuristics, which were tested by running simulations of the game. From these simulations, it becomes apparent that it is wise to give hints about playable cards and discard cards only if completely sure of their worthlessness. Furthermore, it is good to take some risk in playing cards, but not too much.

As for future research, there are many possible routes to take. The first one would be to continue the theoretical analysis of playability as commenced in Section 3.4. In order to do so, one could start by discarding the rule that the game ends when the stack is empty and note that using this modification of the game, a sequence S in $\mathbb{Z}$ is (k, m)-playable if the sequence $(1, \ldots, k)$ can be read when scanning S once from left to right, where a memory of size $m - 1$ is available to store numbers in. Following this observation, it seems that a permutation $(s_i)_{i=1}^N$ of the sequence $(1, \ldots, k)$ is (k, m)-playable if and only if

$$\#\{s_j \mid j < i, s_j > s_i\} \leqslant m - 1$$

for all $i = 1, \ldots, k$. In words: for any number in the sequence, there may not be more than $m - 1$ higher numbers before it, as no more numbers can be stored. Using $\chi_m(x_1, \ldots, x_k)$ to denote the amount of (k, m)-playable sequences disregarding the game's end, we arrive at the following.

Conjecture 5.1. *For $\chi_m(x_1, \ldots, x_k)$ as defined above, we have*

$$\chi_m(1, 1, \ldots, 1) = m!(m + 1)^{k-m}.$$

Furthermore, note that by not stopping the game when the stack is emptied, multi-player games with hand size 1 and infinitely many passes also seem to become equivalent to single-player games:

Conjecture 5.2. *Let* $L = (n, k, p, 1, S_0, \infty, f_0)$ *be an open game of Hanabi, where the game is not ended by an empty stack. Let* $L' = (n, k, 1, p, S_0, 0, f_0)$*. Then* L *is playable if and only if* L' *is playable.*

Instead of further pursuing a theoretical approach, one might also try to enhance Algorithms 3.29 and 3.36 to allow for calculations on larger stacks. It is for example obvious that a unique card must not be discarded for the remaining stack to be playable. Furthermore, in checking whether we need to hold on to a card in our hand to produce a playable sequence, only the first occurrence of every value in the remaining stack would need to be known. Perhaps these observations would resolve some time and space issues involved in the calculations on playability.

Finally, one could also try to further consider the second main question of the thesis regarding optimal strategies. More heuristics could be tried, for example, or more structural search algorithms could be implemented as described in Section 4.4. In addition to the other improvements and extensions discussed in this section, it might also be interesting to take a look at computer Bridge [15] and see whether methods from this area of study may come in handy when examining Hanabi. Indeed, the implicit information present in the hints of this game slightly resembles the implicit information which is given in the auction phase of the game of Bridge, see also [16].

References

[1] R. Bjarnason, P. Tadepalli and A. Fern, Searching Solitaire in Real Time, International Computer Games Association Journal 30 (2007), 131–142.

[2] R. Merris, Combinatoris, 2nd edition, Wiley Interscience, 2003.

[3] B. Modreski, Hanabi English Translation, BoardGameGeek, `http://boardgamegeek.com/filepage/59655/hanabi-english-translation` [retrieved 08.07.2015].

[4] National Institute of Standards and Technology, NIST Digital Library of Mathematical Functions, Section 26.4, `http://dlmf.nist.gov/26.4` [retrieved 08.07.2015].

[5] J.J. Rotman, A First Course in Abstract Algebra, 3rd edition, Prentice Hall, 2005.

[6] Anonymous Users, Stackexchange, Sequences that Contain Subsequence 1,2,3, `http://math.stackexchange.com/questions/1215764/sequences-that-contain-subsequence-1-2-3` [retrieved 08.07.2015].

[7] Anonymous Users, Stackexchange, Probability of Getting A to K on Single Scan of Shuffled Deck, `http://math.stackexchange.com/questions/1248381/probability-of-getting-a-to-k-on-single-scan-of-shuffled-deck` [retrieved 09.07.2015].

[8] S.M. Ruiz, An Algebraic Identity Leading to Wilson's Theorem, Mathematical Gazette 80 (1996), 579–582.

[9] A. Levitin, Introduction to The Design and Analysis of Algorithms, 3rd edition, Pearson Education, 2012.

[10] G.M.J-B. Chaslot, Monte-Carlo Tree Search, PhD thesis, Maastricht University, the Netherlands 2010, `https://project.dke.maastrichtuniversity.nl/games/files/phd/Chaslot_thesis.pdf` [retrieved 08.07.2015].

[11] S. Gelly, M. Schoenauer, M. Sebag, O. Teytaud, L. Kocsis, D. Silver and C. Szepesvári, The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions, Communications of the ACM 55 (2012), 106–113.

[12] D.E. Knuth and R.W. Moore, An Analysis of Alpha-Beta Pruning, Artificial Intelligence 6 (1975), 293–326.

[13] M. Clerc, Particle Swarm Optimization, ISTE, 2006.

[14] E. Aarts and J. Korst, Simulated Annealing and Boltzmann Machines, Wiley Interscience, 1989.

[15] S.J.J. Smith, D. Nau, and T. Throop, Computer Bridge a Big Win for AI Planning, AI Magazine 19 (1998), 93–106.

[16] T. Ando and T. Uehara, Reasoning by Agents in Computer Bridge Bidding, Proceedings Second International Conference Computers and Games (CG2000), 2000, 346–364.

Appendix A

$(x_1, x_2, x_3, x_4, x_5)$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$(1, 1, 1, 1, 6)$	0.0250	0.1460	0.2393	0.2571	0.2571
$(1, 1, 1, 2, 5)$	0.0380	0.1995	0.2893	0.3000	0.3000
$(1, 1, 1, 3, 4)$	0.0436	0.2210	0.2963	0.3000	0.3000
$(1, 1, 1, 4, 3)$	0.0436	0.2276	0.2984	0.3000	0.3000
$(1, 1, 1, 5, 2)$	0.0380	0.2180	0.2952	0.3000	0.3000
$(1, 1, 1, 6, 1)$	0.0250	0.1810	0.2821	0.3000	0.3000
$(1, 1, 2, 1, 5)$	0.0380	0.2110	0.3190	0.3357	0.3357
$(1, 1, 2, 2, 4)$	0.0559	0.2755	0.3768	0.3857	0.3857
$(1, 1, 2, 3, 3)$	0.0613	0.2938	0.3821	0.3857	0.3857
$(1, 1, 2, 4, 2)$	0.0559	0.2866	0.3806	0.3857	0.3857
$(1, 1, 2, 5, 1)$	0.0380	0.2435	0.3690	0.3857	0.3857
$(1, 1, 3, 1, 4)$	0.0436	0.2357	0.3399	0.3500	0.3500
$(1, 1, 3, 2, 3)$	0.0613	0.2959	0.3948	0.4000	0.4000
$(1, 1, 3, 3, 2)$	0.0613	0.3006	0.3958	0.4000	0.4000
$(1, 1, 3, 4, 1)$	0.0436	0.2632	0.3891	0.4000	0.4000
$(1, 1, 4, 1, 3)$	0.0436	0.2427	0.3448	0.3500	0.3500
$(1, 1, 4, 2, 2)$	0.0559	0.2889	0.3952	0.4000	0.4000
$(1, 1, 4, 3, 1)$	0.0436	0.2634	0.3919	0.4000	0.4000
$(1, 1, 5, 1, 2)$	0.0380	0.2308	0.3439	0.3500	0.3500
$(1, 1, 5, 2, 1)$	0.0380	0.2450	0.3880	0.4000	0.4000
$(1, 1, 6, 1, 1)$	0.0250	0.1889	0.3321	0.3500	0.3500
$(1, 2, 1, 1, 5)$	0.0380	0.2181	0.3429	0.3655	0.3655
$(1, 2, 1, 2, 4)$	0.0559	0.2858	0.4083	0.4214	0.4214
$(1, 2, 1, 3, 3)$	0.0613	0.3051	0.4156	0.4214	0.4214
$(1, 2, 1, 4, 2)$	0.0560	0.2975	0.4139	0.4214	0.4214
$(1, 2, 1, 5, 1)$	0.0380	0.2519	0.3988	0.4214	0.4214
$(1, 2, 2, 1, 4)$	0.0559	0.2988	0.4492	0.4690	0.4690
$(1, 2, 2, 2, 3)$	0.0785	0.3715	0.5199	0.5333	0.5333
$(1, 2, 2, 3, 2)$	0.0785	0.3755	0.5212	0.5333	0.5333
$(1, 2, 2, 4, 1)$	0.0559	0.3285	0.5087	0.5333	0.5333
$(1, 2, 3, 1, 3)$	0.0613	0.3202	0.4704	0.4857	0.4857
$(1, 2, 3, 2, 2)$	0.0785	0.3768	0.5340	0.5500	0.5500
$(1, 2, 3, 3, 1)$	0.0613	0.3445	0.5271	0.5500	0.5500
$(1, 2, 4, 1, 2)$	0.0559	0.3111	0.4688	0.4857	0.4857
$(1, 2, 4, 2, 1)$	0.0559	0.3292	0.5228	0.5500	0.5500
$(1, 2, 5, 1, 1)$	0.0380	0.2607	0.4525	0.4857	0.4857
$(1, 3, 1, 1, 4)$	0.0436	0.2461	0.3811	0.4012	0.4012
$(1, 3, 1, 2, 3)$	0.0613	0.3100	0.4444	0.4595	0.4595
$(1, 3, 1, 3, 2)$	0.0613	0.3149	0.4455	0.4595	0.4595
$(1, 3, 1, 4, 1)$	0.0436	0.2744	0.4331	0.4595	0.4595
$(1, 3, 2, 1, 3)$	0.0613	0.3231	0.4866	0.5095	0.5095
$(1, 3, 2, 2, 2)$	0.0785	0.3802	0.5515	0.5762	0.5762
$(1, 3, 2, 3, 1)$	0.0613	0.3474	0.5425	0.5762	0.5762

$(x_1, x_2, x_3, x_4, x_5)$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$(1, 3, 3, 1, 2)$	0.0613	0.3291	0.4995	0.5262	0.5262
$(1, 3, 3, 2, 1)$	0.0613	0.3486	0.5543	0.5929	0.5929
$(1, 3, 4, 1, 1)$	0.0436	0.2845	0.4841	0.5262	0.5262
$(1, 4, 1, 1, 3)$	0.0436	0.2529	0.3875	0.4083	0.4083
$(1, 4, 1, 2, 2)$	0.0559	0.3019	0.4442	0.4667	0.4667
$(1, 4, 1, 3, 1)$	0.0436	0.2743	0.4365	0.4667	0.4667
$(1, 4, 2, 1, 2)$	0.0559	0.3136	0.4858	0.5167	0.5167
$(1, 4, 2, 2, 1)$	0.0559	0.3319	0.5393	0.5833	0.5833
$(1, 4, 3, 1, 1)$	0.0436	0.2845	0.4859	0.5333	0.5333
$(1, 5, 1, 1, 2)$	0.0380	0.2395	0.3841	0.4083	0.4083
$(1, 5, 1, 2, 1)$	0.0380	0.2541	0.4291	0.4667	0.4667
$(1, 5, 2, 1, 1)$	0.0380	0.2624	0.4657	0.5167	0.5167
$(1, 6, 1, 1, 1)$	0.0250	0.1948	0.3643	0.4083	0.4083
$(2, 1, 1, 1, 5)$	0.0380	0.2288	0.3641	0.3903	0.3903
$(2, 1, 1, 2, 4)$	0.0559	0.3012	0.4352	0.4512	0.4512
$(2, 1, 1, 3, 3)$	0.0613	0.3222	0.4435	0.4512	0.4512
$(2, 1, 1, 4, 2)$	0.0559	0.3144	0.4417	0.4512	0.4512
$(2, 1, 1, 5, 1)$	0.0380	0.2655	0.4250	0.4512	0.4512
$(2, 1, 2, 1, 4)$	0.0559	0.3151	0.4787	0.5028	0.5028
$(2, 1, 2, 2, 3)$	0.0785	0.3935	0.5556	0.5730	0.5730
$(2, 1, 2, 3, 2)$	0.0785	0.3982	0.5570	0.5730	0.5730
$(2, 1, 2, 4, 1)$	0.0559	0.3475	0.5431	0.5730	0.5730
$(2, 1, 3, 1, 3)$	0.0613	0.3384	0.5017	0.5214	0.5214
$(2, 1, 3, 2, 2)$	0.0785	0.3997	0.5706	0.5917	0.5917
$(2, 1, 3, 3, 1)$	0.0613	0.3646	0.5627	0.5917	0.5917
$(2, 1, 4, 1, 2)$	0.0559	0.3287	0.4998	0.5214	0.5214
$(2, 1, 4, 2, 1)$	0.0559	0.3481	0.5578	0.5917	0.5917
$(2, 1, 5, 1, 1)$	0.0380	0.2747	0.4820	0.5214	0.5214
$(2, 2, 1, 1, 4)$	0.0559	0.3267	0.5142	0.5464	0.5464
$(2, 2, 1, 2, 3)$	0.0785	0.4098	0.5989	0.6246	0.6246
$(2, 2, 1, 3, 2)$	0.0785	0.4151	0.6003	0.6246	0.6246
$(2, 2, 1, 4, 1)$	0.0559	0.3611	0.5823	0.6246	0.6246
$(2, 2, 2, 1, 3)$	0.0785	0.4264	0.6537	0.6921	0.6921
$(2, 2, 2, 2, 2)$	0.1004	0.5004	0.7382	0.7804	0.7810
$(2, 2, 2, 3, 1)$	0.0785	0.4565	0.7248	0.7802	0.7810
$(2, 2, 3, 1, 2)$	0.0785	0.4333	0.6680	0.7127	0.7135
$(2, 2, 3, 2, 1)$	0.0785	0.4582	0.7378	0.8008	0.8024
$(2, 2, 4, 1, 1)$	0.0559	0.3740	0.6452	0.7119	0.7135
$(2, 3, 1, 1, 3)$	0.0613	0.3560	0.5553	0.5940	0.5940
$(2, 3, 1, 2, 2)$	0.0785	0.4222	0.6317	0.6742	0.6750
$(2, 3, 1, 3, 1)$	0.0613	0.3838	0.6191	0.6738	0.6750
$(2, 3, 2, 1, 2)$	0.0785	0.4393	0.6864	0.7437	0.7452
$(2, 3, 2, 2, 1)$	0.0785	0.4644	0.7569	0.8337	0.8369

$(x_1, x_2, x_3, x_4, x_5)$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$(2, 3, 3, 1, 1)$	0.0613	0.3995	0.6831	0.7631	0.7667
$(2, 4, 1, 1, 2)$	0.0559	0.3461	0.5533	0.6008	0.6024
$(2, 4, 1, 2, 1)$	0.0559	0.3666	0.6144	0.6802	0.6833
$(2, 4, 2, 1, 1)$	0.0559	0.3798	0.6647	0.7488	0.7536
$(2, 5, 1, 1, 1)$	0.0380	0.2878	0.5278	0.5984	0.6024
$(3, 1, 1, 1, 4)$	0.0436	0.2673	0.4228	0.4502	0.4502
$(3, 1, 1, 2, 3)$	0.0613	0.3385	0.4944	0.5171	0.5171
$(3, 1, 1, 3, 2)$	0.0613	0.3441	0.4954	0.5171	0.5171
$(3, 1, 1, 4, 1)$	0.0436	0.2986	0.4810	0.5171	0.5171
$(3, 1, 2, 1, 3)$	0.0613	0.3525	0.5407	0.5742	0.5742
$(3, 1, 2, 2, 2)$	0.0785	0.4168	0.6131	0.6500	0.6508
$(3, 1, 2, 3, 1)$	0.0613	0.3793	0.6023	0.6496	0.6508
$(3, 1, 3, 1, 2)$	0.0613	0.3592	0.5540	0.5925	0.5937
$(3, 1, 3, 2, 1)$	0.0613	0.3804	0.6145	0.6679	0.6702
$(3, 1, 4, 1, 1)$	0.0436	0.3089	0.5356	0.5913	0.5937
$(3, 2, 1, 1, 3)$	0.0613	0.3678	0.5787	0.6230	0.6230
$(3, 2, 1, 2, 2)$	0.0785	0.4366	0.6576	0.7063	0.7079
$(3, 2, 1, 3, 1)$	0.0613	0.3963	0.6443	0.7056	0.7079
$(3, 2, 2, 1, 2)$	0.0785	0.4540	0.7142	0.7786	0.7817
$(3, 2, 2, 2, 1)$	0.0785	0.4798	0.7868	0.8719	0.8778
$(3, 2, 3, 1, 1)$	0.0613	0.4123	0.7098	0.7975	0.8040
$(3, 3, 1, 1, 2)$	0.0613	0.3813	0.6103	0.6694	0.6730
$(3, 3, 1, 2, 1)$	0.0613	0.4038	0.6756	0.7543	0.7607
$(3, 3, 2, 1, 1)$	0.0613	0.4192	0.7299	0.8280	0.8373
$(3, 4, 1, 1, 1)$	0.0436	0.3277	0.5906	0.6733	0.6813
$(4, 1, 1, 1, 3)$	0.0436	0.2788	0.4391	0.4716	0.4716
$(4, 1, 1, 2, 2)$	0.0559	0.3342	0.5029	0.5381	0.5397
$(4, 1, 1, 3, 1)$	0.0436	0.3025	0.4935	0.5373	0.5397
$(4, 1, 2, 1, 2)$	0.0559	0.3469	0.5490	0.5948	0.5980
$(4, 1, 2, 2, 1)$	0.0559	0.3671	0.6085	0.6704	0.6758
$(4, 1, 3, 1, 1)$	0.0436	0.3135	0.5471	0.6117	0.6175
$(4, 2, 1, 1, 2)$	0.0559	0.3633	0.5880	0.6433	0.6480
$(4, 2, 1, 2, 1)$	0.0559	0.3847	0.6517	0.7265	0.7341
$(4, 2, 2, 1, 1)$	0.0559	0.3985	0.7044	0.7987	0.8091
$(4, 3, 1, 1, 1)$	0.0436	0.3335	0.6030	0.6880	0.6980
$(5, 1, 1, 1, 2)$	0.0380	0.2654	0.4371	0.4724	0.4764
$(5, 1, 1, 2, 1)$	0.0380	0.2816	0.4873	0.5389	0.5444
$(5, 1, 2, 1, 1)$	0.0380	0.2907	0.5278	0.5956	0.6028
$(5, 2, 1, 1, 1)$	0.0380	0.3043	0.5647	0.6440	0.6528
$(6, 1, 1, 1, 1)$	0.0250	0.2157	0.4133	0.4716	0.4764

Appendix B

$\omega_s = 1$, $\omega_d = 1$, $\omega_h = 1$:

		ω_p	Hint rule ρ_h			
			1	2	3	4
Discard rule ρ_d	1	1.0	$(5.5056, 14, 0)$	$(12.5932, 22, 2)$	$(13.1048, 21, 4)$	$(10.7417, 20, 3)$
		0.9	$(5.5299, 15, 0)$	$(12.6483, 21, 3)$	$(13.2534, 21, 4)$	$(10.8474, 20, 2)$
		0.8	$(5.8733, 15, 0)$	$(13.1144, 21, 4)$	$(13.9214, 21, 4)$	$(11.3789, 21, 3)$
		0.7	$(6.2900, 16, 0)$	$(13.4682, 22, 3)$	$(14.5354, 22, 6)$	$(11.9149, 22, 3)$
		0.6	$(7.0842, 17, 0)$	$(13.9480, 23, 3)$	$(15.2845, 23, 5)$	$(12.6737, 23, 3)$
		0.5	$(7.2655, 20, 0)$	$(13.7624, 24, 4)$	$(15.1522, 24, 5)$	$(12.5262, 24, 2)$
		0.4	$(6.9327, 20, 0)$	$(13.4423, 24, 3)$	$(14.7761, 24, 4)$	$(12.2090, 24, 3)$
		0.3	$(6.3639, 20, 0)$	$(12, 9593, 23, 3)$	$(13.7684, 24, 3)$	$(11.3396, 25, 2)$
		0.2	$(5.9188, 16, 0)$	$(12.5794, 22, 2)$	$(13.0553, 21, 4)$	$(10.7026, 22, 2)$
		0.1	$(5.9151, 17, 0)$	$(12.5987, 22, 3)$	$(13.0654, 22, 4)$	$(10.7144, 21, 2)$
		0.0	$(5.9053, 17, 0)$	$(12.5707, 22, 4)$	$(13.0284, 23, 4)$	$(10.7025, 21, 2)$
	2	1.0	$(5.7713, 16, 0)$	$(12.6614, 22, 1)$	$(12.8471, 21, 3)$	$(10.7758, 21, 2)$
		0.9	$(5.7916, 16, 0)$	$(12.8029, 21, 1)$	$(13.0706, 21, 2)$	$(10.8978, 20, 3)$
		0.8	$(6.1781, 17, 0)$	$(13.3883, 22, 2)$	$(13.8633, 23, 3)$	$(11.4885, 22, 3)$
		0.7	$(6.6139, 18, 0)$	$(13.8258, 22, 2)$	$(14.5393, 23, 4)$	$(12.0299, 22, 2)$
		0.6	$(7.3921, 19, 0)$	$(14.3677, 23, 3)$	$(15.4075, 23, 4)$	$(12.7149, 24, 2)$
		0.5	$(7.5030, 19, 0)$	$(14.1546, 25, 2)$	$(15.3015, 25, 3)$	$(12.5837, 24, 2)$
		0.4	$(7.1368, 20, 0)$	$(13.7957, 24, 2)$	$(14.9218, 24, 3)$	$(12.2222, 24, 3)$
		0.3	$(6.5577, 19, 0)$	$(13.2201, 24, 1)$	$(13.7877, 24, 2)$	$(11.4213, 25, 2)$
		0.2	$(6.1372, 17, 0)$	$(12.8190, 23, 1)$	$(12.9537, 24, 2)$	$(10.7822, 21, 1)$
		0.1	$(6.1284, 19, 0)$	$(12.8493, 22, 1)$	$(12.9318, 22, 2)$	$(10.8034, 21, 2)$
		0.0	$(6.1131, 17, 0)$	$(12.7790, 22, 2)$	$(12.9477, 23, 3)$	$(10.8185, 20, 2)$
	3	1.0	$(4.9605, 14, 0)$	$(12.2293, 20, 5)$	$(12.6672, 21, 5)$	$(10.2653, 20, 3)$
		0.9	$(4.9973, 15, 0)$	$(12.3364, 20, 5)$	$(12.7818, 21, 5)$	$(10.3829, 20, 3)$
		0.8	$(5.3725, 15, 0)$	$(12.7421, 21, 4)$	$(13.5218, 21, 6)$	$(10.9540, 20, 4)$
		0.7	$(5.8092, 16, 0)$	$(13.1218, 22, 5)$	$(14.1131, 21, 6)$	$(11.5054, 21, 4)$
		0.6	$(6.6410, 16, 0)$	$(13.5421, 22, 5)$	$(14.9423, 23, 7)$	$(12.2981, 22, 3)$
		0.5	$(6.8717, 18, 0)$	$(13.4395, 23, 5)$	$(14.8424, 24, 5)$	$(12.2740, 24, 3)$
		0.4	$(6.5035, 21, 0)$	$(13.1259, 23, 5)$	$(14.4649, 24, 5)$	$(11.9009, 24, 3)$
		0.3	$(5.8765, 20, 0)$	$(12.6055, 24, 4)$	$(13.3777, 24, 4)$	$(10, 9657, 24, 2)$
		0.2	$(5.3761, 15, 0)$	$(12.2265, 22, 4)$	$(12.6381, 22, 4)$	$(10.2441, 21, 3)$
		0.1	$(5.3695, 16, 0)$	$(12.2303, 21, 4)$	$(12.6497, 22, 3)$	$(10.2747, 21, 3)$
		0.0	$(5.3771, 17, 0)$	$(12.2155, 21, 4)$	$(12.6133, 22, 4)$	$(10.2451, 21, 2)$
	4	1.0	$(4.8698, 14, 0)$	$(12.0331, 20, 3)$	$(12.3029, 21, 4)$	$(10.3384, 20, 3)$
		0.9	$(4.8925, 14, 0)$	$(12.1448, 21, 2)$	$(12.4649, 21, 3)$	$(10.4248, 19, 2)$
		0.8	$(5.3105, 15, 0)$	$(12.6284, 21, 3)$	$(13.2326, 21, 4)$	$(10.9843, 20, 3)$
		0.7	$(5.7849, 15, 0)$	$(13.0691, 22, 4)$	$(13.9301, 22, 5)$	$(11.5031, 21, 3)$
		0.6	$(6.6534, 17, 0)$	$(13.6847, 23, 4)$	$(14.8472, 23, 5)$	$(12.2824, 23, 2)$
		0.5	$(6.8430, 20, 0)$	$(13.5090, 24, 2)$	$(14.7366, 24, 3)$	$(12.2453, 25, 3)$
		0.4	$(6.4443, 20, 0)$	$(13.1023, 24, 3)$	$(14.3649, 24, 5)$	$(11.8658, 24, 3)$
		0.3	$(5.7565, 21, 0)$	$(12.5732, 23, 2)$	$(13.2425, 24, 3)$	$(10.9986, 24, 2)$
		0.2	$(5.2733, 17, 0)$	$(12.1734, 22, 2)$	$(12.4184, 23, 3)$	$(10.4085, 21, 2)$
		0.1	$(5.2553, 18, 0)$	$(12.1610, 21, 2)$	$(12.4047, 21, 3)$	$(10.4037, 21, 2)$
		0.0	$(5.2510, 16, 0)$	$(12.1498, 21, 2)$	$(12.4119, 22, 2)$	$(10.4001, 20, 2)$

$\omega_s = 0$, $\omega_d = 1$, $\omega_h = 1$:

		ω_p	Hint rule ρ_h	
			2	3
Discard rule ρ_d	1	1.0	$(12.5932, 22, 2)$	$(13.1048, 21, 4)$
		0.9	$(12.6357, 22, 0)$	$(13.2652, 22, 0)$
		0.8	$(13.0565, 22, 0)$	$(13.8906, 22, 0)$
		0.7	$(13.1259, 22, 0)$	$(14.1914, 22, 0)$
		0.6	$(10.2284, 23, 0)$	$(11.5433, 24, 0)$
		0.5	$(2.2244, 24, 0)$	$(3.1172, 25, 0)$
		0.4	$(0.4891, 24, 0)$	$(0.9215, 24, 0)$
		0.3	$(0.0090, 24, 0)$	$(0.0171, 24, 0)$
		0.2	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.1	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.0	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
	2	1.0	$(12.6614, 22, 1)$	$(12.8471, 21, 3)$
		0.9	$(12.8046, 21, 1)$	$(13.0843, 21, 0)$
		0.8	$(13.3537, 23, 0)$	$(13.8488, 24, 0)$
		0.7	$(13.5718, 23, 0)$	$(14.3005, 23, 0)$
		0.6	$(11.1379, 24, 0)$	$(11.8696, 24, 0)$
		0.5	$(2.8141, 24, 0)$	$(3.0940, 25, 0)$
		0.4	$(0.6115, 24, 0)$	$(0.8301, 24, 0)$
		0.3	$(0.0063, 24, 0)$	$(0.0164, 25, 0)$
		0.2	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.1	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.0	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
	3	1.0	$(12.2293, 20, 5)$	$(12.6672, 21, 5)$
		0.9	$(12.3350, 20, 0)$	$(12.7957, 21, 0)$
		0.8	$(12.6677, 21, 0)$	$(13.4759, 21, 0)$
		0.7	$(12.7402, 22, 0)$	$(13.7811, 22, 0)$
		0.6	$(9.7045, 22, 0)$	$(11.1711, 23, 0)$
		0.5	$(2.0157, 23, 0)$	$(3.1852, 24, 0)$
		0.4	$(0.4338, 25, 0)$	$(1.0019, 24, 0)$
		0.3	$(0.0047, 23, 0)$	$(0.0184, 24, 0)$
		0.2	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.1	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.0	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
	4	1.0	$(12.0331, 20, 3)$	$(12.3029, 21, 4)$
		0.9	$(12.1350, 21, 0)$	$(12.4855, 21, 0)$
		0.8	$(12.5778, 21, 0)$	$(13.2245, 21, 0)$
		0.7	$(12.7804, 23, 0)$	$(13.6581, 23, 0)$
		0.6	$(10.1671, 23, 0)$	$(11.3533, 23, 0)$
		0.5	$(2.0241, 24, 0)$	$(2.9102, 24, 0)$
		0.4	$(0.4320, 24, 0)$	$(0.8405, 25, 0)$
		0.3	$(0.0051, 23, 0)$	$(0.0150, 25, 0)$
		0.2	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.1	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$
		0.0	$(0.0000, 0, 0)$	$(0.0000, 0, 0)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 2,\ \rho_h = 3$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(14.5390, 23, 4)$	$(15.3862, 24, 4)$	$(15.3318, 24, 4)$
0.9	$(14.5594, 22, 3)$	$(15.3884, 24, 5)$	$(15.3284, 24, 4)$
0.8	$(14.4814, 22, 4)$	$(15.2814, 24, 5)$	$(15.2054, 24, 4)$
0.7	$(14.2941, 23, 4)$	$(15.0831, 23, 4)$	$(14.9624, 24, 3)$
0.6	$(13.6756, 22, 3)$	$(14.3426, 23, 4)$	$(14.3206, 23, 4)$
0.5	$(12.1948, 23, 2)$	$(12.8148, 23, 3)$	$(13.0637, 23, 2)$
0.4	$(11.3456, 22, 2)$	$(12.0016, 22, 2)$	$(12.3321, 22, 2)$
0.3	$(9.7709, 21, 2)$	$(10.6428, 21, 2)$	$(10.7501, 21, 2)$
0.2	$(7.9446, 20, 1)$	$(8.3093, 20, 2)$	$(8.1522, 20, 1)$
0.1	$(4.8198, 18, 1)$	$(5.0473, 18, 1)$	$(5.2114, 18, 1)$
0.0	$(0.0813, 4, 0)$	$(0.3801, 6, 0)$	$(0.6603, 8, 0)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 1,\ \rho_h = 3$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(14.5550, 23, 5)$	$(15.2535, 24, 5)$	$(15.1971, 24, 4)$
0.95	$(14.5309, 22, 4)$	$(15.2778, 24, 5)$	$(15.1435, 25, 4)$
0.9	$(14.5387, 22, 4)$	$(12.2650, 23, 5)$	$(15.1811, 24, 4)$
0.85	$(14.5185, 22, 5)$	$(15.2621, 24, 5)$	$(15.1280, 24, 4)$
0.8	$(14.4364, 22, 4)$	$(15.1761, 24, 5)$	$(15.0411, 24, 4)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 2,\ \rho_h = 1$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(6.6103, 18, 0)$	$(7.3725, 18, 0)$	$(7.4781, 20, 0)$
0.95	$(6.6056, 17, 0)$	$(7.3683, 19, 0)$	$(7.4998, 20, 0)$
0.9	$(6.6221, 17, 0)$	$(7.3631, 18, 0)$	$(7.4811, 19, 0)$
0.85	$(6.6005, 17, 0)$	$(7.3633, 19, 0)$	$(7.5021, 19, 0)$
0.8	$(6.6011, 17, 0)$	$(7.3803, 18, 0)$	$(7.4820, 20, 0)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 2,\ \rho_h = 2$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(13.8341, 22, 3)$	$(14.3602, 24, 3)$	$(14.1649, 23, 3)$
0.95	$(13.8383, 22, 2)$	$(14.3718, 23, 4)$	$(14.1760, 25, 3)$
0.9	$(13.8115, 23, 2)$	$(14.3773, 23, 1)$	$(14.1685, 24, 2)$
0.85	$(13.8432, 23, 2)$	$(14.3529, 23, 2)$	$(14.1613, 24, 2)$
0.8	$(13.7957, 23, 3)$	$(14.3137, 23, 3)$	$(14.0983, 24, 3)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 2,\ \rho_h = 3$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(14.5390, 23, 4)$	$(15.3863, 24, 4)$	$(15.3318, 24, 4)$
0.95	$(14.5586, 23, 5)$	$(15.3981, 24, 4)$	$(15.3107, 25, 4)$
0.9	$(14.5161, 23, 5)$	$(15.3645, 24, 4)$	$(15.2956, 25, 5)$
0.85	$(14.5235, 23, 4)$	$(15.3767, 23, 5)$	$(15.3075, 24, 2)$
0.8	$(14.4894, 22, 3)$	$(15.2937, 24, 4)$	$(15.2230, 24, 4)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 2,\ \rho_h = 4$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(11.9910, 22, 2)$	$(12.6953, 24, 3)$	$(12.5871, 24, 3)$
0.95	$(11.9902, 22, 3)$	$(12.7082, 24, 3)$	$(12.5830, 24, 4)$
0.9	$(11.9883, 21, 2)$	$(12.7127, 23, 3)$	$(12.5692, 24, 2)$
0.85	$(12.0060, 22, 3)$	$(12.7479, 23, 3)$	$(12.6224, 24, 2)$
0.8	$(11.9960, 22, 2)$	$(12.7048, 24, 3)$	$(12.5931, 24, 3)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 3,\ \rho_h = 3$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(14.1064, 22, 6)$	$(14.9089, 23, 6)$	$(14.8009, 24, 6)$
0.95	$(14.1364, 22, 6)$	$(14.9155, 24, 7)$	$(14.8373, 24, 6)$
0.9	$(14.1256, 22, 7)$	$(14.8801, 23, 6)$	$(14.8504, 24, 6)$
0.85	$(14.1319, 22, 7)$	$(14.8752, 23, 6)$	$(14.8108, 24, 6)$
0.8	$(14.0663, 21, 6)$	$(14.7715, 24, 6)$	$(14.7296, 23, 6)$

$\omega_s = 1,\ \omega_h = 1,\ \rho_d = 4,\ \rho_h = 3$:

ω_d	ω_p		
	0.7	0.6	0.5
1.0	$(13.9381, 22, 5)$	$(14.8180, 23, 4)$	$(14.7612, 25, 4)$
0.95	$(13.9775, 22, 5)$	$(14.8221, 23, 5)$	$(14.7841, 25, 4)$
0.9	$(13.9486, 22, 5)$	$(14.8346, 24, 5)$	$(14.7270, 24, 4)$
0.85	$(13.9107, 23, 4)$	$(14.8005, 23, 4)$	$(14.7613, 24, 4)$
0.8	$(13.8918, 22, 4)$	$(14.7641, 23, 4)$	$(14.6528, 25, 4)$

$\omega_p = 0.6,\ \omega_s = 1,\ \omega_d = 1$:

ω_h	$\rho_d = \rho_h = 2$	$\rho_d = 2, \rho_h = 3$	$\rho_d = 3, \rho_h = 2$	$\rho_d = \rho_h = 3$
1.0	$(14.3602, 24, 3)$	$(15.3863, 24, 4)$	$(13.5553, 22, 4)$	$(14.9089, 23, 6)$
0.9	$(14.1930, 23, 2)$	$(15.2398, 24, 3)$	$(13.3761, 22, 4)$	$(14.7730, 23, 5)$
0.8	$(13.9194, 23, 1)$	$(14.9892, 23, 3)$	$(13.1068, 22, 4)$	$(14.5573, 23, 5)$
0.7	$(13.3620, 23, 2)$	$(14.5166, 24, 2)$	$(12.4564, 22, 3)$	$(14.1135, 23, 5)$
0.6	$(12.2327, 22, 1)$	$(13.5044, 23, 2)$	$(11.2787, 22, 1)$	$(13.1100, 23, 3)$
0.5	$(10.5193, 22, 0)$	$(11.7572, 23, 2)$	$(9.5636, 20, 1)$	$(11.5150, 22, 2)$
0.4	$(8.5117, 21, 0)$	$(9.7813, 21, 2)$	$(7.7266, 19, 0)$	$(9.7195, 20, 1)$
0.3	$(6.5051, 19, 0)$	$(7.8872, 21, 1)$	$(6.0289, 18, 0)$	$(7.9067, 20, 1)$
0.2	$(4.6148, 16, 0)$	$(6.0293, 17, 0)$	$(4.4217, 16, 0)$	$(6.0844, 16, 0)$
0.1	$(2.6954, 14, 0)$	$(3.8577, 15, 0)$	$(2.7172, 12, 0)$	$(3.9087, 13, 0)$
0.0	$(0.3868, 5, 0)$	$(0.3856, 6, 0)$	$(0.3823, 7, 0)$	$(0.3779, 6, 0)$