A deep dive into Hanabi

Maximilian Keßler

August 15, 2025

Contents

1 stuff to do

Theory – we ignore the clues given, they are simply stalls.

- introduce solitaire and cheating play
- concerete game states: invariants, puzzles ...
- proof that we do not need to bomb at less than 8 clues

2 Introduction

HANABI is a cooperative card-game for 2–5 players. In the default variant, it is played with 50 cards

2.1 Rules

3 Overview of different Models

Definition 3.1 (Card, Deck)**.** Fix constants n, $c \in \mathbb{N}$, which we will refer to as the **maximum rank** of a card and as the **number of colors**, respectively.

A **card** is an ordered pair (k_i, r_i), where $k_i \in \{1, \dots, c\}$ is the **color** of the card and $r_i \in \{1, \dots, n\}$ is the **rank** of the card.

A **deck** is a (finite) sequence $\sigma = (r_1, k_1), (r_2, k_2), \dots, (r_N, k_N)$ of cards such that every card occurs in it at least once, i.e. for every $(r, k) \in \{1, \dots, n\} \times \{1, \dots, c\}$, there is an index $1 \leqslant i \leqslant N$ with $(r_i, k_i) = (r, k)$.

Definition 3.2 (Multiplicity)**.** Given a deck σ, for some card a, we define its **multiplicity** as the number of occurrences of a in σ and denote it $m(a)$.

The maximum number of occurrences of any card in σ is called the **multiplicity** of σ and we denote it by $m := m(\sigma)$.

Definition 3.3 (Stacks)**.** Fix constants n, $c \in \mathbb{N}$ as in Definition 3.1.

For a game state of Hanabi, the **stacks** (or **progress**) is a function $S\colon \{1, \dots, c\} \to \{0, \dots, n\}$ indicating the number of cards played per color. $S(k) = r$ means that the last card played in color k has rank r. In particular, $S(k) = 0$ means that no card has been played.

Given some stacks S, we say that a card (k, r) is **playable** if $S(k) = r - 1$, that is, if (k, r) is the next card of its suit that needs to be played.

The first model that we want to study is the one introduced and studied in [Baf+17]. We use slightly different notation here to fit with the rest of our presentation.

Definition 3.4. An instance of the Singleplayer Solitaire Hanabi problem consists of a deck σ and a **hand size** h. A **game state** consists of

- The stacks $S\colon \{1, \dots, c\} \to \{0, \dots, n\}$
- A set H of at most h cards which are said to be **stored** in hand.
- An index $1 \leqslant i \leqslant N$ indicating the next card of the deck to be drawn.

A **turn** consists of taking one of the following actions:

- *Discarding* the next of the deck, which means setting $i \leftarrow i + 1$.

- If $|H| < h$: *Storing* the next card of the deck, which means setting $H \leftarrow H \cup \{\sigma_i\}$ and $i \leftarrow i + 1$.
- If $\sigma_i = (k_i, r_i)$ is playable: *Playing* the next card of the deck, which means setting $i \leftarrow i + 1$ and $S(k_i) \leftarrow r_i$.
- If a card $(k, r) \in H$ is playable: *Playing* this stored card, which means setting $H \leftarrow H \setminus \{(k, r)\}$ and $S(k) \leftarrow r$.

The game ends if there is no legal action that can be taken. It is said to be **won** if all cards have been played, that is, if $S(k) = n$ for every color k at the end of the game.

The **initial game state** is defined by $S(k) \leftarrow 0$, $H \leftarrow \varnothing$ and $i \leftarrow 1$.

The Singleplayer Solitaire Hanabi problem asks whether there exists a sequence of legal actions that wins this initial game state.

Remark 3.4.1 (Discarding stored cards)**.** Note that this model does not allow discarding cards in hand. This is in fact not necessary, since any card stored in hand and not eventually played could just have been discarded in the first place instead of storing it. This only leads to a smaller number of cards stored in hand, so no further restrictions on other taken actions apply.

So discarding cards in hand is actually not needed and we keep our model simpler by disallowing this in the first place.

Remark 3.4.2 (Partial game states)**.** While this formulation is quite general, it also seems a bit restrictive at first glance, for example, every color has the same number of ranks and it does not allow asking whether a partial game state (i.e a few actions have been taken already) can be won.

However, let us briefly sketch that these are not actually restrictions:

If we want to model some color having only ranks $1, \ldots, m < n$, then we can simply add the cards of ranks $m + 1, \ldots, n$ at the end of the deck and nowhere else. A winning sequence of this instance will have to play up to rank m before reaching this end sequence, and conversely any such sequence can be extended to a winning one by simply playing these cards in order without having to store any of them in hand.

If we want to model some partial gamestate, to model the current stacks, we simply shift the corresponding colors so that the next card played is of rank 1 again and the stacks just end earlier. This is possible by our previous reduction. The current progress in the deck is just modeled by truncating the deck (and removing all cards that are played already). Finally, to model cards that are currently held in hand, we just put them at the start of the deck. Then, we can choose to store them in hand at the start of the game or discard them (which models discarding cards already stored in hand).

Thus, it suffices to study the given model.

Remark 3.4.3 (Decision vs. Computation Problem)**.** Of course, instead of only posing the decision problem, we could ask for such a winning sequence or the certification that none exists.

However, as discussed above, the decision problem also allows asking for partial game states. Thus, given an oracle for the decision problem, we can recover a winning sequence (or assert that none exists) by checking for every legal action whether they yield a winnable next game state. Since any winning game state has at least one such action, we can take that action and recurse. This will then recover a winning sequence.

4 The Singleplayer Solitaire Hanabi Problem

5 Winrates: Theoretical boundaries

5.1 Classes of known losses

5.1.1 Pace losses

Loosely speaking, a **Pace loss** occurs when the team runs out of time to play all cards needed to win. Recall that after the last card is drawn from the deck, the game ends after p more turns, where p is the number of players in the game.

Lemma 5.1. Consider a HANABI state with p players and n cards left in the draw pile. Then, at most $n + p$ cards can be played before the end of the game.

Proof. Each time a card is played, a card is drawn from the deck, with the only exception being the last round of the game. Hence, for all but p cards played, there must be a card in the draw pile to be drawn. So $\#\{\text{played cards}\} - p \leqslant n$, proofing the lemma. □

This motivates the following definition:

Definition 5.2. Consider a state of HANABI with p players and n cards left in the draw pile. Let s be the number of cards that still need to be played on the stacks for the game to be won. Then, we define the **pace** of this state as $p+n-s$.

We immediately get:

Corollary 5.3. Consider a state of HANABI and let P be the current pace. If P is negative, the state is infeasible.

Proof. From the definition, we know that $p + n - s < 0$, hence $p + n < s$. But by Lemma 5.1, no more than $p + n$ cards can still be played, which is strictly less than c. □

Remark 5.3.4. A slight variant of this, which is often easier to think about, is the case where $P = 0$: The inequality from the proof of Corollary 5.3 will be sharp in this case, so every newly drawn card will need to be the result of a card being played. In particular, there are no more discards possible.

As a special case, if you reach pace 0 and there are no more playable cards in any player's hand, the game state is lost as well: You cannot take (successful)

play actions until after discarding, but discarding puts you to negative pace.

So far, we have only looked at specific game states. We now want to draw our attention to decks, in particular decks where.

Definition 5.4 (Selection with exempt subsets)**.** Let M be a set of n elements and $k \in \mathbb{N}$. Let $A_1, \dots, A_r$ and $B_1, \dots, B_s$ be pairwise-disjoint subsets of M, where $|A_i| = 2$ and $|B_i| = 3$. Then, define

$$\mathcal{S}(n,k,r,s) := |\{N \subseteq M \mid |N| = k, A_i \nsubseteq N, B_i \nsubseteq N\}|$$

as the number of k-element subsets of M not containing any of the subsets A_i or B_i entirely.

Remark 5.4.5. Note that the definition of $\mathcal{S}(n,k,r,s)$ is actually independent of the specific subsets A_i and B_i, since all of them are pairwise-disjoint. Hence, only the number of them matters and above definition is well-defined.

Theorem 5.5. Let n, k, r, $s \in \mathbb{N}$, where $k \leqslant n$, $2r + 3s \leqslant n$. Then,

$$\mathcal{S}(n,k,r,s) = \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \left((-1)^i \sum_{\substack{\mu \\ 2\mu} \; \substack{+\nu = i \\ +3\nu \leqslant k}} \binom{r}{\mu} \binom{s}{\nu} \binom{n-2\mu-3\nu}{k-2\mu-3\nu} \right).$$

Proof. We will use the inclusion-exclusion formula. Let $A_1, \dots, A_r$, $B_1, \dots, B_s$ be as in Definition 5.4 and consider

$$\tilde{A}_i := \{N \subseteq M \mid |N| = k, A_i \subseteq N\}, \qquad \tilde{B}_i := \{N \subseteq M \mid |N| = k, B_i \subseteq N\}.$$

Then, $\mathcal{S}(n,k,r,s)$ will be the number of elements in

$$S := \{N \subseteq M \mid |N| = k\} \setminus \left(\underbrace{\bigcup_{i=1}^{r} \tilde{A}_i \cup \bigcup_{i=1}^{s} \tilde{B}_i}_{=:X} \right).$$

To calculate $|X|$, we can now use inclusion-exclusion. □

Let us focus on a 3-player case first.

4 cards left in the deck: This means 7 cards can still be played and hence need to be unaccessible. Since there are only 4 cards, we can have at most 2 summands, and the only possible sums are $4 + 4$ and $4 + 3$, corresponding to the cases of 2's being blocked in 2 colors, or one 2 and one 3 being blocked.

For double 2's, this happens for a fraction of

$$\frac{\binom{5}{2} \cdot 4!}{50^{(4)}} = \frac{10 * 24}{50 * 49 * 48 * 47} = \frac{1}{23,030} \approx 4.34 \cdot 10^{-6}$$

of the decks. For 2's and 3's, we get

$$\frac{5 \cdot 4 \cdot 4!}{50^{(4)}} = \frac{1}{11515} \approx 8.68 * 10^{-6}$$

5 cards left in the deck: Possible sums are $8 = 5+3$, $8 = 4+4+0$ and $8 = 4+3+1$. These result in probabilities of

$$\begin{aligned}
\frac{5 \cdot 4 \cdot 5!}{50^{(5)}} &= \frac{1}{105938} \approx 0.94 \cdot 10^{-6} \\
\frac{\binom{5}{2} \cdot (50-4-3) \cdot 5!}{50^{(5)}} &= \frac{43}{211876} \approx 20.29 \cdot 10^{-6} \\
\frac{5 \cdot 4 \cdot 3 \cdot 5!}{50^{(5)}} &= \frac{3}{105938} \approx 2.83 \cdot 10^{-6}
\end{aligned}$$

6 cards left in the deck Sums are $9 = 5 + 4 + 0$, $9 = 5 + 3 + 1$, $9 = 4 + 4 + 1 + 0$, $9 = 4 + 3 + 2$, resulting in

$$\begin{aligned}
\frac{5 \cdot 4 \cdot (50-5-3) \cdot 6!}{50^{(6)}} &= \frac{2}{37835} \approx 5.29 \cdot 10^{-6} \\
\frac{5 \cdot 4 \cdot 3 \cdot 6!}{50^{(6)}} &= \frac{1}{264845} \approx 0.38 \cdot 10^{-6} \\
\frac{\binom{5}{2} \cdot 3 \cdot (50-5-2) \cdot 6!}{50^{(6)}} &= \frac{43}{529690} \approx 8.12 \cdot 10^{-6} \\
\frac{5 \cdot 4 \cdot 3 \cdot 6!}{50^{(6)}} &= \frac{1}{264845} \approx 0.38 \cdot 10^{-6}
\end{aligned}$$

remaining cards in deck	pace	#occurrences
4	-1	4
4	0	8
5	-1	3
5	0	19
6	-1	2
6	0	21
7	0	9
8	0	10
9	-1	1
9	0	8
10	-1	1
10	0	2
11	0	6
12	0	2
13	-1	1
14	-1	1
14	0	3
15	0	2
18	0	1
19	0	1

Table 5.1: Frequency of pace losses in 3-player Hanabi. Occurrences counted in a sample of $100,000$ randomly generated decks. Decks are counted multiple times if they have nonpositive pace at multiple cuts.

Bibliography

[Baf+17] Jean-Francois Baffier et al. *Hanabi is NP-hard, Even for Cheaters who Look at Their Cards*. 2017. arXiv: 1603.01911 [cs.DM].