

# Query Complexity of Mastermind Variants

Aaron Berger   Christopher Chute   Matthew Stone

December 26, 2015

## Abstract

We study variants of Mastermind, a popular board game in which the objective is sequence reconstruction. In this two-player game, the codemaker constructs a hidden sequence  $H = (h_1, h_2, \dots, h_n)$  of colors selected from an alphabet  $\mathcal{A} = \{1, 2, \dots, k\}$  (i.e.,  $h_i \in \mathcal{A}$  for all  $i \in \{1, 2, \dots, n\}$ ). The game then proceeds in turns, each of which consists of two parts: in turn  $t$ , the codebreaker first submits a query sequence  $Q_t = (q_1, q_2, \dots, q_n)$  with  $q_i \in \mathcal{A}$  for all  $i$ , and second receives feedback  $\Delta(Q_t, H)$ , where  $\Delta$  is a function of distance between two  $n$ -sequences. The game terminates when  $Q_t = H$ , and the codebreaker seeks to end the game in as few turns as possible. Throughout we let  $f(n, k)$  denote the smallest integer such that the codebreaker can determine any  $H$  in  $f(n, k)$  turns.[1] We prove three main results: First, we show that given a set  $S_t$  of sequences which is known to contain  $H$ , there always exists a query whose feedback implicitly produces a smaller set  $S_{t+1}$  such that  $H \in S_{t+1}$  and  $|S_{t+1}| \leq (1 - 1/(nk))|S_t|$ . Second, when  $k = n$  and  $h_i \neq h_j$  for all  $i \neq j$ , we prove that  $f(n, k) \geq n - \log \log(n)$  for all sufficiently large  $n$ . Third, when feedback is not present in the game, we show that there exists a constant  $c > 0$  such that  $f(n, k) \geq c \cdot n \log(k)$ .

## 1 Introduction

Variants of Mastermind are a family of two-player games centered around reconstruction of a hidden sequence. In all variants of the game, one player is given the role of “codemaker,” and the other is denoted the “codebreaker.” The codemaker begins the game by constructing a hidden sequence  $H = (h_1, h_2, \dots, h_n)$  where each component is selected from an alphabet  $\mathcal{A} = \{1, 2, \dots, k\}$  of  $k$  colors (that is,  $h_i \in \mathcal{A}$  for all  $i \in \{1, 2, \dots, n\}$ ). The goal of the codebreaker is to uniquely determine the hidden sequence  $H$  through a series of queries, which are submissions of vectors of the form  $Q_t = (q_1, q_2, \dots, q_n)$ . The codebreaker always seeks to determine  $H$  with as few queries as possible, however the nature of these queries, the feedback received after a query, and the restrictions on  $H$  differ between variants.

The variants which we will study are differentiated by settings of the tuple  $(n, k, \Delta, R, A)$ . These parameters are defined as follows:

- (i) *(n) Length of Sequence.* The parameter  $n$  denotes the length of the hidden sequence  $H$  created by the codemaker, hence  $H = (h_1, h_2, \dots, h_n)$ . The codebreaker is also required to submit query vectors of length  $n$ , so the  $t^{\text{th}}$  query vector takes the form  $Q_t = (q_1, q_2, \dots, q_n)$ .
- (ii) *(k) Size of Alphabet.* This parameter determines the number of possible values for components of  $H$  and  $Q_t$ . Each game is accompanied by an alphabet  $\mathcal{A} = \{1, 2, \dots, k\}$  from which the components of  $H$  and  $Q_t$  are selected. That is,  $h_i \in \{1, 2, \dots, k\}$  and  $q_i \in \{1, 2, \dots, k\}$ .
- (iii) *( $\Delta$ ) Distance Function.* On the  $t^{\text{th}}$  turn of a game, the codebreaker submits a query sequence  $Q_t = (q_1, q_2, \dots, q_n)$ . The codemaker then gives feedback  $\Delta(Q_t, H)$ , which is roughly a measure of distance between  $Q_t$  and  $H$ . The information yielded by  $\Delta(Q_t, H)$  may be used to guide the choice of  $Q_{t+1}$ , the next sequence to be guessed. Hence the choice of distance function affects the codebreaker’s ability to make an informed query on the following turn. We study the following distance functions:

- a. “*Black hits and white hits.*” Let  $Q_t$  and  $H$  be as above. The black hits and white hits distance function is defined by  $\Delta(Q_t, H) = (b(Q_t, H), w(Q_t, H))$  where

$$b(Q_t, H) = |\{i \in \mathbb{Z} \mid q_i = h_i, 1 \leq i \leq n\}|, \quad (1)$$

and

$$w(Q_t, H) = \max_{\sigma} b(\sigma(Q_t), H),$$

where  $\sigma$  iterates over all permutations of  $Q_t$ . We note that this is the distance function used in the original game of Mastermind.

- b. “*Black hits only.*” When  $\Delta$  is the black hits only distance function, it is defined by  $\Delta(Q_t, H) = b(Q_t, H)$ , where  $b$  is defined as in equation (1).

- (iv) (*R*) *Repetition.* The parameter  $R$  is a Boolean restriction on the components of  $H$ . If  $R$  is true, we say that the variant game is *with repeats* or that repeats are allowed. In this case, the hidden vector  $H$  may have repeated colors, that is, we allow  $h_i = h_j$  for any  $i, j \in \{1, 2, \dots, n\}$ . When  $R$  is false, we say that the variant is *no repeats*, and we require  $h_i \neq h_j$  when  $i \neq j$ . In particular, when  $R$  is false and  $k = n$ , we have that  $H$  must be a permutation of  $(1, 2, \dots, n)$ , and we refer to this variant as the *Permutation Game*.

- (v) (*A*) *Adaptiveness.* The Boolean parameter  $A$  determines whether the codebreaker receives feedback after each query. If  $A$  is true, we say that the game is *adaptive*. In this case the game consists of two-part turns: on the  $t^{\text{th}}$  turn, the codebreaker first submits a query sequence  $Q_t$ , and then receives feedback  $\Delta(Q_t, H)$ . The codebreaker may use the feedback to inform the query  $Q_{t+1}$  made in turn  $t + 1$ , and the game ends in turn  $s$  if and only if  $Q_s = H$ .

When  $A$  is false, we say that the game is *non-adaptive*. In this case the codebreaker submits  $m$  queries  $Q_1, Q_2, \dots, Q_m$  all at once (where the codebreaker chooses  $m$ ). The codemaker then reports a feedback vector of the form  $(\Delta(Q_1, H), \Delta(Q_2, H), \dots, \Delta(Q_m, H))$ , after which the codebreaker must submit the final query  $\bar{Q}$ . The codebreaker wins if and only if  $\bar{Q} = H$ .

Throughout we define  $f(n, k, \Delta, R, A)$  to be the smallest integer such that the codebreaker can determine any hidden sequence  $H$  in  $f(n, k, \Delta, R, A)$  queries during a game with the corresponding assignment of  $n, k, \Delta, R$ , and  $A$ . For example, Donald Knuth’s result that the original game of Mastermind (four positions, six colors, black and white hits, with repeats, adaptive) can always be solved in five turns is equivalently stated as  $f(4, 6, \Delta = (b, w), R = T, A = T) = 4$ . We will write simply  $f(n, k)$  when the context is clear.

We prove the following three main results:

1. **Theorem 1.** Let  $A = T$ , let  $\Delta$  be the black hits distance function, and let  $n, k$ , and  $R$  be fixed. Given a set  $S_t$  of sequences which is known to contain  $H$ , there exists a query whose feedback produces a smaller set  $S_{t+1}$  such that  $H \in S_{t+1}$  and  $|S_{t+1}| \leq (1 - 1/(nk))|S_t|$ .
2. **Theorem 2.** Consider the Permutation Game defined by  $n = k$  and  $R = F$ . Let  $\Delta = b$  (black hits only distance function) and let  $A$  be fixed. Then for all sufficiently large  $n$ , we have

$$f(n, k = n, \Delta = b, R = F, A) \geq n - \log \log(n).$$

3. **Theorem 3.** Consider any non-adaptive Mastermind variant, defined by  $A = F$ . Let  $n, k, \Delta$  and  $R$  be fixed. Then there exists a constant  $c > 0$  such that

$$f(n, k, \Delta, R, A = F) \geq c \cdot n \log(k).$$

## References

- [1] CHVÁTAL, V.