Solving Static Permutation Mastermind using O(n log n) Queries

Mastermind is a well known two-player board game, whose commercial version was introduced in the 1970's by Mordecai Meirowitz. The game goes as follows: the first player, named *Codemaker*, secretly chooses a codeword made up of 4 pegs, each of 6 possible colours. The second player, *Codebreaker*, then tries to guess this codeword in as few queries as possible. A query also consists of 4 pegs of 6 possible colours, and for each query Codebreaker receives the number of pegs they guessed correctly (black pegs) as well as the number pegs of the correct colour, but not at the right position (white pegs).

In 1977, Knuth [7] used a minimax argument to prove that five queries were sufficient to find any codeword. Numerous variants of this game have been studied in the literature. Some of the most common variants are:

(i) the number of pegs n and the number of colours k may be arbitrary;

(ii) in the *black-peg only* setting, Codebreaker only receives the black pegs as answer to their queries;

(iii) when $k \geqslant n$, one may or may not allow *repetitions* of colours in the codeword and the queries;

(iv) in the *static* setting, Codebreaker needs to decide all queries in advance, before receiving any answer. In the *adaptive* setting, queries may be adapted depending on the answers to previous queries.

One of the first problem of this type to receive attention is *Coin-Weighing*, a problem introduced in 1960 by Shapiro and Fine [9], which can be proven equivalent to n-peg, 2-colour Mastermind. In 1963, Erdős and Rényi [4] showed via a probabilistic argument that $O(n/\log n)$ queries were sufficient to crack the code. Building up on their idea, Chvátal [1] showed the existence of a strategy for black-peg only Mastermind with repetition, using $O(n \log k / \log n)$ queries when $k \leqslant n^{1-\varepsilon}$. This matches the information-theoretic lower bound in both the adaptive and static cases up to a constant factor.

Determining the minimum number of queries for static, black-peg only Mastermind with repetitions is part of the wider problem of determining the *metric dimension of graphs*. In a recent paper, Jiang and Polyanskii [6] determined, up to lower order terms, the *metric dimension* of large powers of fixed graphs. Concerning Mastermind, their result imply that for any *constant* number of colours k, the query complexity is $(2 + o(1))n/\log_k n$.

For larger k's, determining the correct query-complexity turned out to be more delicate. When $k = n$, Chvátal's approach can easily be adapted to find a bound of $O(n \log n)$, a factor $\log n$ away from the information-theoretic lower bound of $\Omega(n)$ (attributed to Duchet, see [2]). Doerr, Doerr, Spöhel and Thomas [2] showed that this was the correct bound (up to constant factors) in the static case. They also presented an adaptive strategy using $O(n \log \log n)$ queries, by giving a reduction to a sequence of Coin-Weighings problems. The gap to the lower bound was finally closed in a recent paper by Martinsson and Su [8], who showed that by first performing a sequence of $O(n)$ simple "pre-processing" queries, Coin-Weighing schemes can be modified to match the information-theoretic lower bound for Mastermind. In the adaptive case, optimal bounds for $k > n$ can be deduced from this bound, we refer to [2] and [8] for the details.

In the case of Permutation Mastermind we assume $k = n$ and allow no repetitions of colors, neither in the codeword nor in the queries. In this setting, one usually views the codeword and the queries as permutations of $[n]$, thus the name. It is trivial to see that black-peg only, and black- and white-peg settings are equivalent in this case. Restricting the codeword to permutations obviously makes it easier for Codebreaker to win, on the other hand this same restriction for queries rules out many of the usual strategies. In the adaptive setting, the best lower- and upper-bounds currently, of order $\Omega(n)$ and $O(n \log n)$, are both due to El Ouali, Glazik, Sauerland and Srivastav [3]. Where the exact query-complexity lies inside this $\Theta(\log n)$ gap is still an open question.

Recently, Glazik, Jäger, Schiemann and Srivastav [5] shed some light on *Static Permutation Mastermind*. Adapting an argument of Doerr, Doerr, Spöhel and Thomas [2], they proved that any strategy requires at least $\Omega(n \log n)$ queries. Additionally, they show that there exists a strategy using $O(n^{1.525})$ queries and ask whether it is possible to find

a strategy matching their lower bound. In this paper we answer this question positively.

Theorem 1. *For* Static Permutation Mastermind*, there exists a set of $O(n \log n)$ queries from which any codeword can be recovered.*

Our proof uses the probabilistic method. We show that with positive probability, a random sequence of $28n \log n$ queries chosen uniformly at random has the property that it can uniquely determine all codewords, hence there are deterministic sequences with this property.

Additionally, one easily observes that *given such a good sequence*, our proof of Theorem 1 can be turned into a simple algorithm which recovers the codeword *efficiently* from such a sequence (by finding one colour at a time in n turns). This contrasts with Chvátal's original result for $k \leqslant n^{1-\varepsilon}$, where no corresponding efficient reconstruction scheme has been presented in the literature, leaving codebreaker to brute-force a codeword that matches all queries. In the case of static classical Mastermind with $k = n$, our approach could be adapted to have an efficient reconstruction algorithm, but we will not elaborate on this further.

For a quick overview of the state of the art on Mastermind, we provide the best known lower- and upper-bounds for the case $k = n$ in Table 1. The rest of this paper is dedicated to the proof of Theorem 1.

	Lower bound	Upper bound
Adaptive Classical	$\Omega(n)$ (see [2])	$O(n)$ [8]
Static Classical	$\Omega(n \log n)$ [2]	$O(n \log n)$ [1]
Adaptive Permutation	$\Omega(n)$ [3]	$O(n \log n)$ [3]
Static Permutation	$\Omega(n \log n)$ [5]	$O(n \log n)$ (our paper)

Table 1: Summary of results for variations of Mastermind with n colours and pegs.

2 Proof of the Theorem

We use standard notations. By log we denote the natural logarithm. For the sake of conciseness we omit floor and ceil signs. Whenever it is needed, we assume that n is large enough. All permutations are assumed to be over $[n] = \{1, \ldots, n\}$.

Our proof of Theorem 1 is probabilistic and in certain ways resembles the original proof of Chvátal. Consider a set of queries Q of size $q(n)$ chosen independently and uniformly at random from all permissible queries. If we could show that it allows to decode a randomly chosen codeword with probability $1 - o(1/n!)$, then a simple union bound argument allows us to conclude that there has to *exist* such a set of queries of size $q(n)$ that allows to decode *all* codewords.

Now, as it turns out, a success probability $1 - o(1/n!)$ in this case is too optimistic. For instance, with probability $1/poly(n)$ a collection of $O(n \log n)$ random queries will never query colors 1 or 2 in positions 1 or 2, in which case, distinguishing permutations

beginning with 12 and 21 is impossible. Instead, to get the union bound to follow through, we need a more refined picture of what "bad events" could cause the recovery to fail. This is made formal in Lemma 2; first, we introduce some definitions.

Given a set $I \subseteq [n]$, we say that $c \in [n]^I$ is a *colouring* of I. We say that it is a *valid colouring* of I if $c(i) \neq c(j)$ for all $i \neq j$, in particular c may be viewed as the restriction of a permutation to I. Given a permutation σ and a colouring c, we say that σ is a 0-*query* for c if $\sigma(i) \neq c(i)$ for all $i \in I$. Given a permutation σ and two colourings c_1, c_2, we say that σ *discriminates* c_1 *from* c_2 on I if σ is a 0-query for c_1 but not for c_2. The following lemma is the core of our argument.

Lemma 2. *There exists a set Q of $28n \log n$ permutations such that the following holds. For any $I \subseteq [n]$, any valid colouring v and any colouring c on I such that $v(i) \neq c(i)$ for all $i \in I$, there exists some $q \in Q$ which discriminates v from c.*

We emphasise that in the above lemma c can be *any* colouring while v needs be a valid one. We defer the proof of this lemma to the end of this section and now prove Theorem 1.

Proof of Theorem 1. Let Q be the set as in Lemma 2. Our strategy to recover any codeword c_0 is to find the colours one by one until we have found them all. We let I be the set of pegs for which we have *not* found the correct colour; at the beginning $I = [n]$.

At any step when $|I| \geqslant 1$, because we know the correct colours outside of I, we can compute for each query $q \in Q$ how many positions c_0 and q colour identically in I. Let $Q_I = \{q_1, \ldots, q_l\} \subseteq Q$ be the set of 0-queries for c_0 on I. We claim that there must exist some $i \in I$ such that $\{c_0(i), q_1(i), \ldots, q_l(i)\} = [n]$. Indeed, if this were not the case, then there would exist some colouring c (which need not be valid) such that $c(i) \notin \{c_0(i), q_1(i), \ldots, q_l(i)\}$ for all $i \in I$. However, this would imply that no $q \in Q_I$ (and hence, in Q) discriminates c_0 (or rather its restriction to I) from c on I, and this contradicts the definition of Q. In consequence, such an $i \in I$ exists, and we can recover $c_0(i)$: it is the unique element of $[n] \setminus \{q_1(i), \ldots, q_l(i)\}$. We remove this i from I and continue until $I = \varnothing$. $\square$

All that remains is now to prove Lemma 2.

Proof of Lemma 2. The proof is probabilistic: we choose independently and uniformly at random a set of $28n \log n$ queries and show that with positive probability, this set satisfies the conditions.

To do this, we wish to apply union-bound over all (I, v, c). A rough application will not work: indeed there are roughly $n^{(2+o(1))n}$ choices of triple (I, v, c), but the probability of failing to discriminate a certain triple may be as large as $1/poly(n)$ if I is small. To solve this problem, we partition the set of triples (I, v, c) depending on the size of I: when $k = |I|$ is small, there are fewer choices for (I, v, c) so it is fine if the probability of discriminating is smaller.

Concretely, assume that for any fixed triple (I, v, c) as in the statement, a random colouring has probability at least $|I|/7n$ of discriminating v from c. Then the probability

that none of the $28n \log n$ random queries discriminate is at most

$$(1 - |I|/7n)^{28n \log n} \leqslant e^{-4|I| \log n} = n^{-4|I|}.$$

When $|I| = k$, a crude upper bound for the choice of (I, v, c) is n^{3k} in total. Hence by union bound, the probability that there exists a triple (I, v, c) with $I \neq \varnothing$ for which no colouring is discriminating is at most

$$\sum_{k=1}^{n} \sum_{|I|=k} \sum_{v,c} n^{-4|I|} \leqslant \sum_{k=1}^{n} n^{3k} n^{-4k} < 1,$$

whenever $n \geqslant 2$.

Therefore, all we need to do is prove that, indeed, for any fixed (I, v, c) as above, a uniformly random permutation has probability at least $|I|/7n$ of discriminating. To bound this, we denote by S_i the set of those permutations σ which are 0-queries for v and such that $\sigma(i) = c(i)$; we now want to show that the size of $\bigcup_{i \in I} S_i$ is at least $|I|(n-1)!/7$. Using inclusion-exclusion, we get

$$\left| \bigcup_{i \in I} S_i \right| \geqslant \sum_{i \in I} |S_i| - \sum_{i \neq j \in I} |S_i \cap S_j|. \tag{1}$$

To estimate this quantity we shall compute the sizes of S_i and $S_i \cap S_j$. We express them in terms of $A(n, |I|)$ which we define as the number of 0-queries of v on I. If we let $m = |I|$, write $\mathfrak{S}_n$ for the set of all permutations on $[n]$ and T_i for the set of permutations σ such that $\sigma(i) = v(i)$, then a simple application of the inclusion-exclusion principle gives

$$A(n, m) = \left| \mathfrak{S}_n \setminus \bigcup_{i \in I} T_i \right| = \sum_{k=0}^{m} \left((-1)^k \binom{m}{k} (n-k)! \right).$$

Claim 3. *For all $m \leqslant n$ we have*

$$n!/3 \leqslant A(n, m) \leqslant n!.$$

Proof. Since $A(n, m)$ is the size of a subset of permutations of $[n]$, the upper bound is trivial. For the lower bound recall that the expression of $A(n, m)$ stems from an inclusion-exclusion argument. As it is well known, that inclusion-exclusion alternatively over- resp. underestimates we get

$$A(n, m) \geqslant \sum_{k=0}^{3} (-1)^k \binom{m}{k} (n-k)!.$$

By spelling out these terms and grouping them appropriately we get $A(n, m) \geqslant (1 - \frac{m}{n})n! + \frac{m(m-1)}{6n(n-1)} \left(3 - \frac{m-2}{n-2} \right) n!$, from which the claimed bound follows easily by observing that the last bracket is at least two and the remaining terms are monotonous in m. $\qquad \square$

Using the above, we now find expressions for the sizes of $|S_i|, |S_i \cap S_j|$.

Claim 4. *For all $i \neq j \in I$, we have*

(i) $|S_i| \geqslant (1 - o(1))A(n, |I|)/n$;

(ii) $|S_i \cap S_j| \leqslant (1 + o(1))A(n, |I|)/n(n - 1)$.

Proof. We prove this claim by appropriate double counting arguments. For a permutation σ that is a 0-query for v we denote by ℓ the index such that $\sigma(\ell) = c(i)$. If $\ell \notin I$ or $\sigma(i) \neq v(\ell)$ then the permutation that is obtained from σ by swapping the colours at positions i and ℓ belongs to S_i. (Note that this case includes the case $i = \ell$, as then σ belongs to S_i already.) As there exist only $n(n - 2)!$ permutations for which $\sigma(i) = v(\ell)$, we thus have

$$A(n, |I|) - o(n!) \leqslant n|S_i|.$$

For (ii), we first note that $S_i \cap S_j = \varnothing$ when $c(i) = c(j)$, thus in this case the claim holds trivially. In the following we thus assume $c(i) \neq c(j)$. Consider an arbitrary $\sigma \in S_i \cap S_j$. For some (arbitrary) indices ℓ_1 and ℓ_2, perform the following operation: swap the colours at position i and ℓ_1 and at positions j and ℓ_2. When do we get a 0-query for v? Clearly, a sufficient condition is to ensure that ℓ_1 and ℓ_2 are different from i and j, the colours of $\sigma(\ell_1)$ and $\sigma(\ell_1)$ do not coincide with $v(i)$ resp. $v(j)$, and – for $\ell_1, \ell_2 \in I$ – the colors of $\sigma(i)$ and $\sigma(j)$ does not coincide with that of $v(\ell_1)$ resp. $v(\ell_2)$. Clearly, we thus have at least $(n - 4)(n - 5)$ proper choices for ℓ_1 and ℓ_2. Observe in addition that for each permutation σ' that is obtained by such a double swap of i, ℓ_1 and j, ℓ_2, there is a unique (σ, ℓ_1, ℓ_2) from which it can be obtained: ℓ_1 (resp. ℓ_2) is the index such that $\sigma'(\ell_1) = c(i)$ (resp. $\sigma'(\ell_2) = c(j)$) and σ is obtained from σ' by swapping the colours of i, j with those of ℓ_1, ℓ_2 respectively. Thus we have

$$(n - 4)(n - 5)|S_i \cap S_j| \leqslant A(n, |I|).$$

The formulas as in (i) and (ii) follow immediately from the above and the bounds of Claim 3, where we catch the change to $n(n - 1)$ in the $o(\cdot)$-notation. $\square$

Putting the expressions of $|S_i|, |S_i \cap S_j|$ from Claim 4 into (1) gives

$$\begin{aligned}
\left| \bigcup_{i \in I} S_i \right| &\geqslant \sum_{i \in I} |S_i| - \sum_{i \neq j \in I} |S_i \cap S_j| \\
&\geqslant |I| \frac{A(n, |I|)}{n} - \binom{I}{2} \frac{A(n, |I|)}{n(n - 1)} + o(|I|(n - 1)!) \\
&= \frac{|I|}{n} A(n, |I|) \left(1 - \frac{|I| - 1}{2(n - 1)} \right) + o(|I|(n - 1)!) \\
&\geqslant (1/6 + o(1))|I|(n - 1)! \qquad\qquad \text{by Claim 3.}
\end{aligned}$$

For a choice of n large enough, this is at least $|I|(n - 1)!/7$ as claimed. $\square$

References

[1] V. Chvátal. Mastermind. *Combinatorica*, 3(3-4):325–329, 1983.

[2] B. Doerr, C. Doerr, R. Spöhel, and H. Thomas. Playing mastermind with many colors. *Journal of the ACM (JACM)*, 63(5):1–23, 2016.

[3] M. El Ouali, C. Glazik, V. Sauerland, and A. Srivastav. On the query complexity of black-peg ab-mastermind. *Games*, 9(1):2, 2018.

[4] P. Erdős and A. Rényi. On two problems of information theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 8:229–243, 1963.

[5] C. Glazik, G. Jäger, J. Schiemann, and A. Srivastav. Bounds for the static permutation mastermind game. *Discrete Mathematics*, 344(3):112253, 2021.

[6] Z. Jiang and N. Polyanskii. On the metric dimension of cartesian powers of a graph. *Journal of Combinatorial Theory, Series A*, 165:1–14, 2019.

[7] D. E. Knuth. The computer as master mind. *Journal of Recreational Mathematics*, 9(1):1–6, 1976.

[8] A. Martinsson and P. Su. Mastermind with a linear number of queries. arXiv:2011.05921, 2020.

[9] H. S. Shapiro and N. Fine. E1399. *The American Mathematical Monthly*, 67(7):697–698, 1960.