Static Mastermind

Wayne Goddard
Department of Computer Science
University of Natal, Durban

ABSTRACT. Static mastermind is like normal mastermind, except that the codebreaker must supply at one go a list of questions (candidate codes), the answers to which must uniquely determine the secret code. We confirm the minimum size list for some small values. Then we solve the game for up to 4 positions. In particular we show that for k sufficiently large the minimium size of a list for 4 positions and k colours is $k-1$.

1 Introduction

In the original version of the game Mastermind (trademark), there are two players, CodeBreaker and CodeSetter. The CodeSetter creates a secret code of 4 pegs, each peg drawn from the same palette of 6 colours. The CodeBreaker must infer the secret code by asking a series of questions. Each question is itself a candidate code, and the response is two integers called Black and White. The value of Black indicates in how many positions there is exact agreement, and the value of White indicates how many colours are correct but in the wrong position. Another way to put this is that the sum of Black and White is the maximum number of blacks over all permutations of the guess code. In the actual game, the response is given in the form of little pegs.

There is now an extensive literature on the game. This includes (a) papers on exact values (e.g. [7] which determines the maximum number of guessed required for the original 6-colour 4-position game and [6] which determines the optimal expected number of guesses for the original game); (b) asymptotics (e.g. [2]); (c) computer strategies based on calculation (e.g. [8]); and (d) computer programs using artificial intelligence ideas (e.g. [4]).

A few authors have considered the question of what happens if it is required that all questions be asked at one go, and then the responses must be enough to uniquely determine the code. In line with Chvátal [2], we denote the minimum number of codes needed $g(n,k)$ for n positions and

JCMCC 47 (2003), pp. 225-236

k colours. Recently, Bogomolny and Greenwell [1, 5] called this variation ***static mastermind***, and calculated the value for a few small cases and provided some upper bounds in a few more cases.

In this paper we completely solve static mastermind for at most 3 positions, and for 4 positions and sufficiently many colours. In particular, we show that for k sufficiently large, $g(4,k) = k-1$. By computer search we also find lower bounds to match the bounds of [1] in almost all cases. In particular we confirm that $g(4,6) = 6$ (the original version).

2 Small Values

The folowing table contains the known values for small numbers of positions and colours. The cases marked with a star were solved/given by Bogomolny and Greenwell [1]. The cases marked with a circle are an upper bound from [1] with a new matching lower bound. The remaining few entries are new.

Colours \ positions	2	3	4	5	6	7	8
2	2*	2*	3*	3°	4°	5°	5°
3	2*	3*	3*	4°	4°	$\leq 5^*$	
4	3*	4*	4°	5°			
5	4*	4*	5°				
6	4*	5°	6°				
7	5*	6°	≤ 7				
8	6°	7	≤ 8				
9	6°	8					
10	7°	9					

Formulas are derived for 2 and 3 positions below. The remaining lower bounds are produced by brute force search. The search space is reduced by exploiting symmetry for the first few choices of codes (as used, for example, in [6]). For example, it is well known that there are up to symmetry only 5 codes in the original 6^4 game; after a code such as 1123 there are 130 codes up to symmetry. Programs available from author. Even with all this symmetry, the original 6^4 game took several days running on a typical desktop PC.

The new upper bounds are the cases of 7 and 8 colours in the four-position game. Example code lists are: 4624, 3756, 2113, 2425, 6334, 7167, 1542; and 1123, 1245, 4172, 5442, 3774, 6826, 7536, 8658.

3 Asymptotics

There is a trivial lower bound for normal Mastermind based on the observation that for n positions and k colours there are k^n codes, and each question can reduce the possibilities by a factor of at most $\binom{n+2}{2}$. The same bound holds for static mastermind:

$$g(n,k) \geq O\left(\frac{n \log k}{\log n}\right).$$

It turned out that Erdős and Renyi [3] solved a problem similar to static mastermind with 2 colours. (It was in the guise of a problem of determing which of a set of coins are counterfeit, by using a scale which tells how many coins are counterfeit in a collection placed on it.) Using similar probabilistic methods, Chvátal [2] determined the order of $g(n,k)$ if the number of positions is large relative to the number of colours. In particular, *for a fixed k*, he showed that

$$g(n,k) \leq O\left(\frac{n \log k}{\log n}\right),$$

which matches the lower bound.

If the number of positions is fixed and the colours grows, then the trivial information-theoretic bound above is of no use. Rather one can see that most colours must be used in most positions so that for fixed n, $g(n,k) \geq O(k)$.

For $n = k$ the trivial lower bound is linear, but we know of no decent upper bound.

4 Two Positions

We use the standard convention of using as our colours the integers from 1 up to k. The following result was independently found by Rosenbaum [9].

Theorem 1 *For k colours and 2 positions, $g(2,k) = \lceil 2k/3 \rceil$.*

In order to find a suitable set of $\lceil 2k/3 \rceil$ guesses, consider the following. Let T be the set of two guesses $\{13, 23\}$. It is easy to see that this set solves the 2-position 3-colour game.

But the set T has the added property that with any number of colours one can determine from just the answers to T how many of each of the colours 1, 2 and 3 are present and in which positions they go, except that

one cannot distinguish between the codes 1x and 11, or between 2x and 22, or between x3 and 33 where x denotes another colour.

It follows that if k is a multiple of 3, that the set $\{13, 23, 46, 56, 79, 89, \ldots\}$ works. To obtain a suitable list with $k = 3r + 1$, take the set for $k - 1$ and add the code kk. For $k = 3r + 2$, take the set for $k - 1$ and add the code (k-1)k. For example, for $k = 5$ use $\{13, 23, 44, 45\}$. It is easily checked that this set works. Indeed, by the property of T described above, it suffices to check that the list works for $k \in \{4, 5\}$.

For a lower bound, let S be a set of guesses that solves static mastermind for 2 positions and k colours. If one colour is absent from S, then every other colour appears at least twice in S (if colour i is absent and colour j appears only once, say in the first position, then one cannot separate ji and jj). Thus $|S| \geq k - 1$.

So suppose every colour appears in S. Call a colour ***thin*** if it appears only once in S; let t denote the number of thin colours. Now we observe that two thin colours cannot occur in the same code: for, if ab is both the unique code containing a and the unique code containing b, then one cannot separate aa and bb. Thus $t \leq |S|$.

On the other hand, by counting the total occurrences of all colours in S, it follows that $2|S| \geq t + 2(k - t) = 2k - t$. Thus

$$2|S| \geq 2k - t \geq 2k - |S|,$$

so that $|S| \geq 2k/3$, as required.

This completes the proof of Theorem 1.

5 Three Positions

We saw earlier that $g(3, k) = k$ for $2 \leq k \leq 4$.

Theorem 2 *For $k \geq 5$ colours and 3 positions, $g(3, k) = k - 1$.*

The key idea in finding a suitable set of guesses is the following. Let T be the set of three guesses $\{112, 322, 313\}$. Note that each number appears in two codes, but three times in total, and the holes with the single colour are different in each case.

We claim: given any code using arbitrary colours, one can determine from just the answers to T how many of each of the colours 1, 2 and 3 is present and in which positions they go, except that one cannot distinguish between the codes 11x and 111, or between x22 and 222, or between 3x3 and 333.

This claim can be proved laboriously be a case analysis, or by letting a computer try T in a game with 4 (or more) colours and seeing which codes are not separated.

Now, by computer check, the set $Q_4 = \{551, 525, 133, 443\}$ works for 5 colours. For $k = 5 + 3r$, we will add r versions of T, with the i^{th} version using colours $5 + 3i - 2$ to $5 + 3i$. For example for $n = 8$, we add 667, 877, 868. By the above claim, given any code, the colours with value 6 or more are completely determined and placed. (To distinguish between the members of an exceptional pair, simpy check whether any other code gets a positive response.) So it suffices to check the system for 8 colours, as 11 or more colours requires the same properties of Q_4. This was done by computer.

Similarly: $Q_4 \cup \{612\}$ works for 6 colours and augmented with r versions of T works for $6+3r$ colours. Similarly: $Q_4 \cup \{612, 747\}$ works for 7 colours and augmented works for $7 + 3r$ colours. (All checked by computer.) So $g(3, k) \leq k - 1$ for $k \geq 5$.

We need to prove a matching lower bound. We use a counting argument. The key is the following idea.

Consider a set S of guesses that solves static mastermind for n positions and k colours. Consider a grid of cells—rows labelled by colours, columns labelled by holes—with a cross in a cell if that colour does not appear in that hole. For a colour i, let $max(i)$ denote the maximum number of times colour i is used in any one guess of S. Let $cross(i)$ denote the number of crosses in row i (that is, the number of positions in which colour i never appears in S). Let $occ(i)$ denote the total number of times i is used in S.

Lemma 1
(a) There cannot be two rows and two columns such that there are crosses in all 4 cells.
(b) If colours i and j have crosses in the same column, then $max(i) + max(j) \geq n$.
(c) In particular, for every pair of colours $cross(i) + cross(j) \leq n$.

PROOF. (a) Suppose crosses appear in the first two columns in both rows 1 and 2; then one cannot distinguish 12... and 21....

(b) Suppose $max(i) + max(j) < n$ and rows i and j have crosses in the first position. Fix any code C with at least $max(i)$ of colour i and at least $max(j)$ of colour j, leaving the first position blank. Consider a secret which is C with the first position either colour i or colour j. The colour of the first position clearly cannot affect the number of blacks for any guess in S. But the total number of pegs for a guess is the sum of the overlaps in each colour, which is at most $max(i)$ for i and at most $max(j)$ for j for any guess

in S, and hence is also not affected by the colour in the first position. That is, one cannot determine the colour in the first position, a contradiction.

(c) If $cross(i) + cross(j) > n$, then there must be a column where both colours have crosses, and so the result follows from part (b). QED

We now establish the lower bound for $n = 3$. The main step is showing

$$\sum_{i=1}^{k} occ(i) \geq 3k - 5.$$

From this it follows that $|S| \geq k - 5/3$ and hence, by integrality, that $|S| \geq k - 1$.

Consider the grid as in the above lemma. Define a colour i as ***deficient*** if $occ(i) < 3$. Define a deficient colour as ***singled*** if it occurs exactly twice in S, each time in a different hole. That is, $occ(i) = 2$ and $max(i) = cross(i) = 1$. Define a deficient colour as ***doubled*** if $max(i) = 2$ (so occurs in unique code of S).

Observation 1 *If two deficient colours have crosses in the same column, then one is doubled and one is not doubled.*

Suppose two deficient colours i and j have crosses in the same position, say the first one. By (b) of above lemma, $max(i) + max(j) \geq 3$; so one of the colours, say i, has $max(i) = 2$, and hence is doubled. If the other colour j is doubled, then one cannot distinguish between iij and iji. This proves the observation.

CASE 1: $occ(i) = 0$ for some colour i. Then by (c) of above lemma, every other colour occurs at least 3 times. So sum of *occ* is at least $3k - 3$.

CASE 2: $cross(i) \leq 1$ for all colours. Then every deficient colour is either singled or doubled. By above observation there are at most 3 doubled and at most 3 singled. Suppose there are three doubled colours, say 1, 2, 3 (necessarily missing different columns). E.g. 11x, 2y2, z33. Then one cannot distinguish between 132 and 213. So there are at most 2 doubled and at most 3 singled, and so sum of *occ* is at least $3k - 5$.

CASE 3: $cross(i) = 2$ for some colour i. By (c) of the above lemma, all other colours have at most one cross and so again all other deficient colours are either singled or doubled. Say i occurs only in the third position.

Let j be a deficient colour. If j misses the third position, then it cannot be doubled (else cannot distinguish between iji and jii). If j misses the first (or second) position, then by (b) of above lemma, j is doubled. So there are at most 3 other deficient colours, and the sum of *occ* is at least $3k - 5$, whence the result.

This completes the proof of Theorem 2.

6 Four Positions

In this section we show that:

Theorem 3 *For 4 positions and k colours, $g(4,k) = k-1$ for all k sufficiently large.*

6.1 Lower bound

Let S be a set of guesses that solves static mastermind for 4 positions and k colours. We establish a lower bound for the total occurrences:

$$\sum_{i=1}^{k} occ(i) \geq 4k - 7, \tag{1}$$

which implies that $|S| > k - 2$.

Define a colour i as ***deficient*** if $occ(i) < 4$. For a deficient colour i, define $sav(i) = n - occ(i)$. The above inequality is implied by

$$\sum_{i=1}^{k} sav(i) \leq 7.$$

Call a deficient colour ***singled*** if it occurs exactly thrice in S, each time in a different code and each time in a different position. That is, $sav(i) = max(i) = cross(i) = 1$. Call a deficient colour ***tripled*** if $max(i) = 3$. Otherwise it is ***clumped***.

By (b) of Lemma 1, there is at most one singled colour that misses any one particular position.

For a deficient colour, we say that it ***confuses*** two positions if any code of S that contains that colour in the one position also has that colour in the other position. Note that any clumped colour confuses at least one pair of positions; a tripled colour confuses the three positions it is in.

It is clear that two colours cannot confuse the same two columns (as one cannot determine which order the two colours appear in in those columns). But we can go further. Define a graph G with vertex set the 4 positions. Two positions are joined by a edge of colour i if that colour confuses those two positions.

Observation 2 *The multigraph G has no non-monochromatic cycle.*

For example, if colour 1 confuses positions 1 and 2, colour 2 confuses positions 2 and 3, and colour 3 confuses positions 1 and 3, then S does not separate 123 and 312.

Now a series of cases.

Case 1: There is a colour with $occ(i) = 0$. Then every other colour has $occ(j) \geq 4$ and so savings is 4.

Case 2: There is a tripled colour. Say code is x111. Clearly no other tripled colour (else repeated confusion). For any clumped colour, to avoid a repeated confusion it must appear in pair involving the first position. But by lack of cycles in G there can be at most one such colour. So savings at most 4 (from the singled colours) plus 2 (from the other deficient colours).

Case 3: There is a colour with $occ(i) = 1$. Say i occurs in the first position. Let j be another deficient colour. If j misses any other position than the first, then by (b) of Lemma 1, it must be tripled, and we've handled that case. If j misses the first position, then if it is singled it is unique, and if doubled then unique by no cycles in G. So savings at most 4.

Case 4: There is colour with $occ(i) = 2$. Say i misses the first two columns. Then at most two singled colours. If $max(i) = 2$, then at most one more edge in G; if $max(i) = 1$ then no other crosses in the missing columns (by lack of tripled colours). In either case savings at most 5.

Case 5: Every colour appears at least 3 times and is never tripled. Then savings from singled colours is at most 4. Each clumped colour contributes one edge to G; so at most 3 clumped colours and total savings at most 7.

6.2 Upper bound

Define a ***design*** as a set of k guesses where:
(i) each guess contains distinct colours,
(ii) each colour occurs in every position,
(iii) no two colours appear together more than once.

Lemma 2 *If $n = 4$ and T is a design, then the code is always determinable except when the code has no repeat, and that case is detectable.*

PROOF. Fix any secret code C. Say it has g colours. The total number of pegs achieved by T is then $4g$. Hence T reveals g.

Define a ***contender*** as a colour all of whose occurrences receive positive response. We say a contender is ***fake*** if it does not occur in C. Each colour in C is a contender. Now, any fake contender appears in 4 different codes,

necessarily each time with a different colour of C (by property (iii) above). So it follows that there cannot be a fake contender if $g < 4$, and so in that case the colours of C are revealed (though not yet their multiplicities).

We say that a colour appears ***solo*** in a position, if there is no other contender in the code of T that contains that colour in that position.

If $g = 1$ we are done. If $g = 2$ then, even ignoring the code where the two colours are together if such a code exists, in each position at least one of the colours appears solo, and whether it gets a Black or a White determines which colour appears in that position.

So assume $g = 3$. One can determine the colour in a position if at least two of the colours appear solo in that position. Further, one can determine the final colour if all but one position is determined (look at the code with the Black peg yet unaccounted for). So we are home unless there are two positions in doubt, and thus the only problem situation is: 12xx and 31xx up to symmetry (and know 2 not in first position and 3 not in second).

Indeed, we are done unless we receive responses of 1 black 1 white for both these guesses. To distinguish between 11xx and 32xx we use the fact that $n \leq 4$. For, in the remaining two columns, there is one colour missing, and so there must one of that in the first two columns. This completes the proof of the lemma. QED

So we can restrict our discussion to secret codes which have all distinct entries. To have success here, one needs considerably more properties of the design. Consider a fake contender. It occurs in 4 codes, each time together with exactly one colour from C. If ever one can determine one colour of C for definite, then one can eliminate that colour and then by the argument used in the above lemma, one can determine the remaining colours.

However, we are looking for a code with $k - 1$ guesses. This is a simple extension.

Lemma 3 *(a) If designs T_1 and T_2 solve static mastermind on k_1 and k_2 colours, then the combination U solves static mastermind on $k_1 + k_2$ colours. (b) For $n = 4$, if a design T solves static mastermind on k colours, then so does the set T' with any one code omitted.*

PROOF. (a) By definition, the combination of two designs (with disjoint colour sets) forms a design. So, by Lemma 2, U works except possibly if all 4 colours are distinct; so assume that. By counting the total responses for T_1 and T_2, one can determine how many colours are from each, say t_1 and t_2. If both t_1 and t_2 are positive, then one can determine which colours are used from each design (by the same argument used in the proof of Lemma 2). Further, WLOG $t_1 \leq 2$ and it is easy to determine the exact positions of these colours.

Finally, pick t_1 new colours from T_2 and consider the secret code with each colour of T_1 replaced by a new colour from T_2. When one adjusts the responses to T_2 accordingly, since T_2 solves static mastermind, one can determine the adjusted secret code; in particular, one can determine the position of the colours from T_2 and so determine the original secret code.

(b) We can reconstruct the response to the missing code! The design T receives a total of 4 blacks over all the guesses. So the number of blacks for the missing code can be inferred. Further, the total number of pegs that T receives is a multiple of 4. So if the total number of pegs for T' is not a multiple of 4, then the missing number is determined. If the total number of pegs for T' is a multiple of 4, then the missing code had either 0 or 4 pegs. It is the latter exactly when the total is 12 and every punctured colour is a contender. (We argued in the proof of Lemma 2 that if the total pegs is not 16 that there can be no fake contenders.) QED

Now we present our suitable design T_{34} on 34 colours. It can be checked by computer that it solves static mastermind on 34 colours. One can easily verify that it is a design. The hefty computation is that if C has separate colours, the colours and their positions are determinable. We also have a design T_{35}. Thus one can construct a suitable set for any k large enough.

1–33–10–11	13–27–24– 4	25–22–15– 3
2–23–17–15	14– 2–34–29	26–21–32–22
3–24–12–31	15–31–29–20	27–11–23–34
4–12– 2–10	16– 4– 1–19	28–25–20–27
5– 3–26–16	17– 9– 5– 6	29– 6–25–13
6–18– 8–23	18– 1– 3– 2	30–26–18–28
7–15– 6–32	19–29–33– 8	31–14–16– 9
8–28–22–24	20–19–11– 5	32–34–28–17
9–10–30–21	21–17–19– 7	33–13–21–14
10–20– 7–18	22– 5–14–30	34–30–13–12
11– 8– 4–25	23–32–31– 1	
12–16–27–33	24– 7– 9–26	

The design T_{34} that solves static mastermind for 4 positions and 34 colours

7 Five or More Positions

Some properties of designs easily generalise to $n = 5$. But the following lemma shows that more codes are required:

Lemma 4 *For $n \geq 5$ there is constant ε_n such that $g(n,k) \geq (1+\varepsilon_n)k - O(1)$.*

PROOF. Let S be a set of guesses that solves static mastermind for n positions and k colours. Consider the colours i with $occ(i) \leq n$. Label such a colour as ***clumped*** if it confuses two or more positions. Define such a colour as ***flat*** if it occurs in each position and each time in a different code. Otherwise it is is ***holey***.

Since two colours cannot confuse the same pair of positions, the number of clumped colours is at most $O(n^2)$. In fact, by Observation 2 it is not hard to argue that the number and total savings from clumped colours is $O(n)$.

Now if a colour is ***holey***, since it has at most n occurrences, does not confuse any positions and is not flat, it must miss a position. We argued before that for two colours sharing a cross position, at least one must have $max \geq n/2$. This precludes two holey colours from having a cross in the same position, since one would be clumped. So there are at most n holey colours, and the total savings from them is at most n.

Now, consider the flat colours. If S contains two codes such as 12... and 31... with all three colours 1, 2 and 3 flat, one cannot distinguish 11123... and 23123.... So for any flat colour, in all the n codes containing it, at most half of the other colours can be flat. It follows that a constant proportion of the positions in the codes of S are from colours with $n+1$ or more occurrences. This proves the lemma. QED

8 Conclusion

We have provided solutions for small cases of static mastermind and provided formulas for the case of 2, 3 or 4 positions. It remains to say something sensible about the general case where the number of positions equals the number of colours.

References

[1] A. Bogomolny and D. Greenwell, Invitation to mastermind, MAA Online, www.maa.org/editorial/knot/mastermind.html, 1999.

[2] V. Chvátal, Mastermind, *Combinatorica* **3** (1983), 325–329.

[3] P. Erdős and C. Rényi, On two problems in information theory, *Magyar Tud. Akad. Mat. Kut. Int. Közl.* **8** (1963), 229-242.

[4] M.M. Flood, Mastermind strategy, *J. Recreational Math.* **18** (1985/86), 194–202.

[5] D. Greenwell, Mastermind, *J. Recreational Math.* **30** (1999–2000), 191–192.

[6] K. Koyama and T.W. Lai, An optimal Masterming strategy, *J. Recreational Math.* **25** (1993), 251–256.

[7] D.E. Knuth, The computer as Mastermind, *J. Recreational Math.* **9** (1976), 1–6.

[8] E. Neuwirth, Some stratgies for Mastermind, *Z. Oper. Res. Ser. A-B* **26** (1982), B257–B278.

[9] G. Rosenbaum, Static Mastermind, http://home.t-online.de/home/guenther.rosenbaum/p0071.htm