Yet another Mastermind strategy

1 author:

Barteld Pieter Kooi
University of Groningen
87 PUBLICATIONS   **2,750** CITATIONS

YET ANOTHER MASTERMIND STRATEGY

Barteld Kooi[1]

University of Groningen, The Netherlands

ABSTRACT

Over the years many easily computable strategies for the game of Mastermind have been proposed. In this paper we present a new strategy (at least to our knowledge) that performs better than the well-known strategies: guess the code that has the highest number of possible answers. It is motivated and compared to four well-known strategies. Some empirical results are presented and discussed.

1. INTRODUCTION

Mastermind[2] is a two-player zero-sum game of imperfect information. First, Player 1 secretly chooses a combination of four pawns drawn from six colours. Then Player 2 can ask up to eight combinations of four pawns as guesses of this secret combination. If she[3] finds the secret combination within eight guesses, Player 2 wins the game, otherwise Player 1 wins the game. Each time Player 2 proposes a guess, she receives an answer by Player 1 that expresses the accuracy of the guess. The answer consists of two numbers: (1) the number of pawns that are both of the right colour and in the right place, and (2) the number of pawns that are of the right colour, but are not in the right place. For example, when the secret combination is "AABB" and the guess is "BBAB" then the answer is: one pawn is in the right place and two pawns have the right colour but are not in the right place (I will abbreviate this answer to (1,2)).

There are many winning strategies for Player 2 guaranteeing that the secret combination is found within eight guesses. There is even a strategy (the "Worst-Case Strategy") that guarantees that the solution is found within five questions (see Table 7). So, there seems little more to say about the game. However, most of the strategies for Mastermind proposed in the literature apply to a slight variation of the game.

A range of papers have been published about Mastermind since the game was first sold in the 1970s. One particular paper by Koyoma and Lai (1993) seems to be the final paper on Mastermind strategies. The authors found the optimal strategy for Player 2 by depth-first search on a supercomputer. With this strategy, the expected number of guesses needed by Player 2 is 4.340. In most of the earlier papers, strategies were put forward which can be calculated more easily. Although these strategies are not optimal, it still seems worthwhile to study them since their limited computational complexity makes these strategies easy to generalize to other settings and variations of the game (e.g., more colours or more pawns). In Section 2, four of the well-known and easily computable Mastermind strategies are presented.

Among the easily computable strategies that have been proposed over the years, one strategy is lacking, namely: make the guess that leads to the highest number of possible answers. This (surprisingly simple) strategy needs only 4.373 guesses on expectation. It is presented and motivated in Section 3. We provide some empirical results in Section 4. The results are discussed in Section 5 and possible explanations are given for the poor behaviour of some of the strategies. In Section 6, we draw conclusions and indicate ideas for further research.

---

[1] Department of Philosophy, University of Groningen, Groningen, The Netherlands. Email: barteld@philos.rug.nl.

[2] The name "Mastermind", the toy, and the distinctive likenesses thereof are the TM properties of Hasbro Inc. The game Mastermind is very similar to the older game "Bulls and Cows", which is played on paper with numbers instead of with coloured pegs. [Editor's note]

[3] For clarity, the first player will be referred to as "he", the second player as "she".

2. FOUR WELL-KNOWN MASTERMIND STRATEGIES

In this section we introduce four of the easily computable strategies that have been proposed over the years: (1) A Simple Strategy, (2) The Worst-Case Strategy, (3) An Expected-Size Strategy, and (4) The Entropy Strategy. After dealing with the first strategy, we provide some insight into the question how to exploit the informativeness of the guesses.

2.1 A Simple Strategy

The first strategy is given by Shapiro (it is also published in Sterling and Shapiro (1994)): it is called "A Simple Strategy". It works as follows: all possible combinations are ordered (usually lexicographically) and the first combination is taken as the first guess. The answer is received. The next guess is the first combination in the ordering that is consistent with the answers given so far. This goes on until the secret combination is cracked. A crucial drawback of this strategy is that it observes the informativeness of the guesses only marginally: Player 2 is only certain that she does not know the answer already, and that is all.

2.2 How to exploit the informativeness of guesses

Before describing the second strategy, we focus on the question how the informativeness of guesses can be exploited. In Mastermind, each guess partitions the set of possible combinations. This can be seen in the following example. Consider a simplified Mastermind game with only two pawns and four colours (A, B, C, and D). The set of possible combinations can be represented as in Figure 1. Figure 2 provides a representation of the answers on the guess (AA), while Figure 3 does so for the guess (DA).

DA	DB	DC	DD
CA	CB	CC	CD
BA	BB	BC	BD
AA	AB	AC	AD

Figure 1: A representation.

$1,0$	$0,0$	$0,0$	$0,0$
$1,0$	$0,0$	$0,0$	$0,0$
$1,0$	$0,0$	$0,0$	$0,0$
$2,0$	$1,0$	$1,0$	$1,0$

Figure 2: Guess AA.

$2,0$	$1,0$	$1,0$	$1,0$
$1,0$	$0,0$	$0,0$	$0,1$
$1,0$	$0,0$	$0,0$	$0,1$
$1,0$	$0,1$	$0,1$	$0,2$

Figure 3: Guess DA.

It is obvious that the guess DA is more informative than the guess AA, but how can this intuition be motivated? The main idea of the three strategies presented below (including our new strategy) is that the choice for a guess is based solely on the *size* of the partitions of the remaining possibilities that a guess generates.

In this respect, in the simplified game above, only two different kinds of guesses are possible at the start of the game: a guess with one colour (e.g., AA), or a guess with two colours (e.g., AB). This is summarized in Table 1. Each number represents the number of combinations for which the answer (in front of the row) would be given to the guess. For example, in Figure 2, there are 9 combinations where the answer is $(0,0)$ for guess AA.

Answer	AA	AB
$0,0$	9	4
$0,1$	0	4
$0,2$	0	1
$1,0$	6	6
$2,0$	1	1

Table 1: Answer combinations for Mastermind with two colours only.

In Table 2, the five different possible partitions for the first guess in a standard Mastermind game (four pawns and six colours) are provided. For example, there are 625 combinations for which the answer is $(0,0)$ when the first guess is $AAAA$. It seems obvious that question $AAAA$ is not a good question, but what more can be said?

Answer	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
0,0	625	256	256	81	16
0,1	0	308	256	276	152
0,2	0	61	96	222	312
0,3	0	0	16	44	136
0,4	0	0	1	2	9
1,0	500	317	256	182	108
1,1	0	156	208	230	252
1,2	0	27	36	84	132
1,3	0	0	0	4	8
2,0	150	123	114	105	96
2,1	0	24	32	40	48
2,2	0	3	4	5	6
3,0	20	20	20	20	20
4,0	1	1	1	1	1

Table 2: Answer combinations for the first guess in Mastermind. For each guess and each answer, the number of combinations that would yield that answer to the guess is given. The positive numbers can also be seen as the sizes of the elements of the partition generated by the guess.

2.3 The Worst-Case Strategy

Assume that Player 2 wants to minimize the number of guesses required to find the secret combination. Then the number of combinations Player 2 considers possible, gives an indication of the number of guesses it will take. The worst thing that can happen to Player 2 in this respect, is that the answer to a guess leaves her with the largest element of the partition (see Table 3). The Worst-Case Strategy, described by Knuth (1976-1977), suggests to pick the guess that minimizes the largest partition element. According to Table 3, Player 2 should guess $AABB$ (with minimum 256).

Guess	Largest element
$AAAA$	625
$AAAB$	317
$AABB$	256
$AABC$	276
$ABCD$	312

Table 3: The sizes of the largest partition elements in Table 2.

2.4 The Expected-Size Strategy

Player 2's decision for a guess might be based on the *expected case* instead of the worst case, when she wants to maximize her *expected payoff*. Therefore, one might consider to look at the expected size of the resulting partition elements. The expected size of a partition element is the probability of getting the answer corresponding to that partition element, multiplied with the size of the partition element. This expectation is defined as follows for the first question in Mastermind. Let $\mathcal{A}$ be the set of possible answers a_i and let $a(x, g)$ be the function that produces the answer for combination x on guess g. The expected size of the resulting partition elements for the *first guess* in Mastermind with $\mathcal{C}$ colours and p pawns is:

$$\sum_{a_i \in \mathcal{A}} \mathbf{P}_g(a_i) . \#(\{x \mid x \in \mathcal{C}^p \wedge a(x, g) = a_i\})$$

where $\mathbf{P}_g(a_i)$ is the probability that the answer to g is a_i and $\#$ stands for the *cardinality* of a set. If one assumes a uniform distribution over all possible combinations, then:

$$\mathbf{P}_g(a_i) = \frac{\#(\{x \mid x \in \mathcal{C}^p \wedge a(x, g) = a_i\})}{\#(C^p)}$$

For example, $\mathbf{P}_{AAAA}(0, 0) = \frac{625}{6^4} = \frac{625}{1296}$. So the expected size $\mathbf{E}(g)$ is:

$$\mathbf{E}(g) = \sum_{a_i \in \mathcal{A}} \frac{\#(\{x \mid x \in \mathcal{C}^p \wedge a(x, g) = a_i\})^2}{\#(\mathcal{C}^p)}$$

For the first guess, the expected sizes are shown in Table 4. From this point of view, Player 2 should pick $AABC$ as the first guess. The expected-size strategy is described by Irving (1978-1979)[4].

Guess g	$\mathbf{E}(g)$
$AAAA$	511.9
$AAAB$	235.9
$AABB$	204.5
$AABC$	185.3
$ABCD$	188.2

Table 4: The expected size of the partition elements in Table 2.

2.5 The Entropy Strategy

There is a measure that gives an ordering on partitions entropy (see Cover and Thomas, 1991). The concept of entropy plays an important role in information theory, since it measures the amount of information contained in messages. Entropy can be used for a Mastermind strategy too, as described by Neuwirth (1982). Such a strategy can be motivated by the following example. Assume we have a guessing game. Player 1 picks a card randomly from a deck of cards. Player 2 has to determine which card Player 1 picked using as few yes/no questions as possible. For instance, if there are eight cards, one needs three questions to determine which card it is (since $\log_2(8) = 3$). The logarithm gives an approximation of the expected number of yes/no questions needed (in fact, it is the limit of the expected number of questions for an infinite number of simultaneously played games).
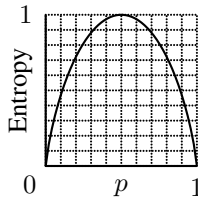

Figure 4: Entropy of a partition with two elements.

Assume we have a partition $V = \{V_1, \ldots, V_m\}$ of a set A. The probability p_i (whether an element of A is in V_i) is $\frac{\#(V_i)}{\#(A)}$ when the distribution is uniform. The expected number of yes/no questions can then be represented as $\sum_{i=1}^{m} p_i \log(\#(V_i))$. Trying to minimize this measure is the same as trying to maximize the entropy, which is defined as $-\sum_{i=1}^{m} p_i \log(p_i)$, since $\log(p_i) = \log\left(\frac{\#(V_i)}{\#(A)}\right) = \log(\#(V_i)) - \log(\#(A))$. Figure 4, displays the entropy for partitions with two elements. The variable for the x-axis is the probability p of one of the elements of the partition, the entropy is given on the y-axis. So, the graph shows the function $-p \log p - (1 - p) \log(1 - p)$. As can be seen in this figure, the highest entropy occurs for the partition in which both parts have equal probability.

Mastermind is very much like the guessing game introduced above, and one can simply calculate the entropies of the first guesses, see Table 5. On the basis of this strategy, Player 2 should start with $ABCD$.

Guess	Entropy
$AAAA$	1.498
$AAAB$	2.693
$AABB$	2.885
$AABC$	3.044
$ABCD$	3.057

Table 5: The entropy of the partitions in Table 2.

---

[4]Irving's paper contains a number of strange (irreproducible) results. First of all he claims that a closer investigation of Knuth's strategy reveals that the total number of guesses required for all 1296 combinations is 5804, whereas it should be 5801, according to our calculations. This can be explained by a minor programming error (the same that we made), but we cannot explain any of his other results. He states that his strategy selects the first two guesses on the basis of the expected number of remaining possibilities and the rest by exhaustive search. When we regard the second guess according to his strategy, my calculations disagree with his on five cases. In four of those he does not take the first of the list available to him. In the other case it is simply wrong. His first guess is $AABC$. If the reply to the guess is (3,0), according to Irving, the next guess should be $FBAC$. (One immediately wonders why not $DBAC$.) According to our calculations, the expected size of the set of remaining possibilities after this guess is 4.7. However, if one guesses $ABCC$ the expected size is 3.6. One difference between these two guesses is that Irving's guess $FBAC$ partitions the remaining possibilities in 8 parts, whereas $ABCC$ partitions the set of remaining possibilities in 7 parts. So it might be the case that he took the average number of remaining possibilities, instead of the expected size, but we still are not able to reproduce his results.

3.   A NEW STRATEGY: MOST-PARTS STRATEGY

In this section we suggest another approach to guessing games. Assume again that Player 2 has to guess which card Player 1 would have drawn randomly from an ordinary deck of cards. Player 2 wins 1\$ if the guess is correct. There are 52 possibilities. Before Player 2 guesses she can ask one yes/no question, which is truthfully answered by Player 1. Which question is best? Intuitively one would think that the question "Is it the Queen of hearts?" is a bad question and that the question "It it a red card?" is a good question. However, all yes/no questions appear to be equally good. This can be seen as follows. Assume the two piles have sizes x and y. Assume that the card is in group x with probability $x/52$. The probability of guessing the right card if it is in this group is then $1/x$. Now assume that the card is in group y with probability $y/52$. The probability of guessing the right card if it is in this group is $1/y$. Hence, the expected gain is:

$$\frac{x}{52} \cdot \frac{1}{x} \cdot 1\$ + \frac{y}{52} \cdot \frac{1}{y} \cdot 1\$ = \frac{2}{52}\$$$

So it does not matter what the sizes of x and y are (as long as they are positive).

This principle can be generalized. Assume there is a set $\mathcal{A}$ and we have to guess what element of $\mathcal{A}$ we are dealing with. We also have to assume that the probability distribution on $\mathcal{A}$ is uniform. Before we guess we can ask a question that can be seen as a partition $V = \{V_i, \ldots, V_n\}$. The probability of guessing correctly, once we learn in which part of V the element is:

$$\sum_{i=1}^{n} \frac{\#(V_i)}{\#(\mathcal{A})} \cdot \frac{1}{\#(V_i)} = \frac{n}{\#(\mathcal{A})}$$

So in these cases, the sizes of the elements of the partition do not matter, only the *size* of the partition, i.e., the number (n) of elements of the partition matters.

This can also be generalized further to games with more than one round, using the principle of complete induction. Assume that the probability of guessing the element of any set $\mathcal{A}$ correctly in a game with r rounds is the number of parts into which partition $\mathcal{A}$ can be divided with r questions, divided by the cardinality of $\mathcal{A}$, where the player's question can depend on the answer to the previous questions (i.e., the induction hypothesis). In case of a game with $r + 1$ rounds, Player 2 can ask $r + 1$ questions, and then has to guess the secret element of $\mathcal{A}$. Let the first question lead to a partition $V = \{V_1 \ldots V_n\}$. Let n_i indicate the number of parts into which V_i can be partitioned with the rest of the questions. Using the induction hypothesis, we infer that when the game is played in r rounds for V_i, the probability of guessing the element of V_i equals n_i divided by the cardinality of V_i. Then the probability of guessing correctly in $r + 1$ rounds for the set $\mathcal{A}$ is:

$$\sum_{i=1}^{m} \frac{\#(V_i)}{\#(\mathcal{A})} \cdot \frac{n_i}{\#(V_i)} = \sum_{i=1}^{m} \frac{n_i}{\#(\mathcal{A})}$$

So, the probability of guessing the element of set $\mathcal{A}$ correctly in a game with $r + 1$ rounds equals the number of parts into which $\mathcal{A}$ can be partitioned with $r + 1$ questions divided by the cardinality of $\mathcal{A}$. By induction one can conclude that this holds for any r and any set $\mathcal{A}$.

This multi-round guessing game is similar to Mastermind, although the questions are guesses themselves. This means that in Mastermind, if one wants to maximize the number of combinations for which one would win in a certain round, one should maximize the number of parts into which the set of all combinations is partitioned in the previous round. It appears not to be feasible to calculate the optimal choice for as many as five rounds, but the idea can be used as a motivation for a strategy. The partitions of the first guess in Table 2 lead to the number of partition elements in Table 6.

Guess	Number
$AAAA$	5
$AAAB$	11
$AABB$	13
$AABC$	14
$ABCD$	14

Table 6: The number of partition elements in Table 2.

So, Player 2 should start with either guess $AABC$ or $ABCD$. In our strategy, first the guesses that maximize the number of parts are selected. Then, if possible, the consistent guesses are selected from these. Finally, lexicographical order is used to select a single guess. So, the first guess of Player 2 in our strategy is $AABC$.

4.   EMPIRICAL RESULTS

Table 7 shows for each of the five strategies for how many combinations the game is won in a particular round of the game. In other words: each strategy produces a game tree and the table shows for every depth of the tree how many leaves (nodes without successors) there are.

Strategy	1	2	3	4	5	6	7	8	9	Total	Expected
					Round						
Simple	1	4	25	108	305	602	196	49	6	7471	5.765
Worst case	1	6	62	533	694	0	0	0	0	5801	4.476
Expected size	1	10	54	645	583	3	0	0	0	5696	4.395
Entropy	1	4	71	612	596	12	0	0	0	5722	4.415
Most parts	1	12	72	635	569	7	0	0	0	5668	4.373

Table 7: Number of combinations for which the strategy produces a win in the specified rounds. The last two columns give the total number of guesses needed and the expected number of guesses.

The table also shows how many guesses are needed in total in the strategy (the sum of the lengths of all the paths from the root of the tree to a leaf) and the expected number of guesses needed (the expected length of a path to a leaf). The last four strategies compare quite favourably to Koyoma and Lai's (1993) result of 4.340.

5.   DISCUSSION

In this section we discuss the empirical results. It seems quite surprising that the Simple Strategy performs so badly with respect to the maximum number of rounds required and the expected number of rounds required. This strategy does not even guarantee that one wins in eight rounds. It seems that the first guess is not a good choice. This can easily be improved by choosing another combination than $AAAA$ for the first guess and let the rest be ordered lexicographically. Starting with $AABB$, for example, gives the results as presented in Table 8, which is considerably better. But it still performs badly in comparison to the other strategies.

Round	1	2	3	4	5	6	7	8	9
Count	1	12	71	253	588	286	78	7	0

Table 8: Number of combinations for which the Simple Strategy starting with $AABB$, produces a win in the specified rounds.

One of the reasons of the bad performance can be explained by the following example. Assume that six combinations remain: $ABAA$, $ABAB$, $ABAF$, $ABDE$, $AEAE$, $AFAE$. In Table 9, for each of these remaining possibilities the answers are shown for the guess in the column.

	$ABAA$	$ABAB$	$ABAF$	$ABDE$	$AEAE$	$AFAE$	$ABFA$
$ABAA$	4,0	3,0	3,0	2,0	2,0	2,0	3,0
$ABAB$	3,0	4,0	3,0	2,0	2,0	2,0	2,1
$ABAF$	3,0	3,0	4,0	2,0	2,0	2,1	2,2
$ABDE$	2,0	2,0	2,0	4,0	2,0	2,0	2,0
$AEAE$	2,0	2,0	2,0	2,0	4,0	3,0	1,1
$AFAE$	2,0	2,0	2,1	2,0	3,0	4,0	1,2

Table 9: Answers for the guesses in the columns for each one of the secret combinations in the rows.

A consistent guess (i.e., a guess that is possibly the secret combination) is not able to distinguish between all six combinations, but guess $ABFA$, which is not one of the six remaining combinations, is able to do so, as can be seen in the table. In this way, both the maximum number of guesses required and the expected number of guesses required can be reduced. In all strategies except the Simple Strategy inconsistent guesses occur.

Another interesting observation is that, although the strategies cannot distinguish between optimal guesses, the actual selection influences the empirical results. When instead of the first, the lexicographically *last* guess is selected by the algorithms in case of a tie between optimal combinations, the results of the strategies change slightly, as can be seen in Table 10.

The Simple Strategy is left out of this table, because these considerations do not affect this

				Round					
Strategy	1	2	3	4	5	6	7	8	9
Worst case	1	8	65	522	696	4	0	0	0
Expected size	1	10	54	646	582	3	0	0	0
Entropy	1	4	70	613	596	12	0	0	0
Most parts	1	12	72	636	568	7	0	0	0

Table 10: Number of combinations for which the strategy produces a win in the specified rounds, using reversed lexicographical ordering.

strategy. The differences are very small. They are largest in case of Knuth's Worst-Case Strategy. In my opinion this simply means that only looking at the partition is not very robust.

Why the results are so very different in the Worst-Case Strategy is because of the following. After the first guess has been answered, the number of ways the set of remaining possibilities can be partitioned in is quite large. As we know, there are only five types of guesses in the initial state. But after the first guess has been answered there are many more. Table 11 shows the number of guesses that can be asked if the first answer is $1, 0$.

First guess	Partitions
$AAAA$	12
$AAAB$	53
$AABB$	34
$AABC$	125
$ABCD$	52

Table 11: Number of partitions after the first guess and answer 1,0.

So in the Worst-Case Strategy, there are already 34 different kinds of partitions possible after the first guess. This strategy only looks at one aspect of these partitions and apparently this is not fine-grained enough to yield a robust strategy. If there are already 34 guesses possible after the first guess, this will even be worse after more guesses.

The Expected-Size Strategy is straightforward, and indeed it requires 6 rounds, but on average it is better than the Worst-Case Strategy.

One of the surprising results is that the Entropy Strategy does so bad, although its motivation seems to be theoretically sound. A possible explanation is that when one calculates the entropy, the base of the logarithm is important when one compares partitions that have a different number of elements. When one compares partitions with the same size, entropy is a good measure, otherwise it is not so good. Perhaps another new strategy could be based on taking entropy where the base of the logarithm depends on the size of the partition.

The Most-Parts Strategy is the best strategy when regarding the expected number of questions. The only problem is that theoretically the number of rounds matters, whereas this is ignored in selecting a guess. From Table 7 it follows that up to rounds 2,3, or 4, the Most-Parts Strategy is better than the other strategies. However, in calculating the next guess this strategy only looks one step ahead. Table 12 gives results for looking two steps ahead.

The numbers in the table represent the number of different answers one could get on a guess, after the initial guess and the initial answer. The total number at the bottom is the total number of parts of the partition that results from

Answer	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
0,0	14	14	14	13	8
0,1	0	14	14	14	13
0,2	0	9	12	14	14
0,3	0	0	7	10	11
0,4	0	0	1	2	4
1,0	13	14	14	14	13
1,1	0	13	14	14	14
1,2	0	7	10	11	11
1,3	0	0	0	4	4
2,0	11	12	12	12	12
2,1	0	9	10	11	9
2,2	0	3	4	4	4
3,0	5	8	8	8	7
4,0	1	1	1	1	1
Total	44	104	121	132	125

Table 12: Number of answers in the second round.

asking two guesses. So if the game consists of three rounds, it is best to start with $AABC$. Unfortunately, looking two steps ahead is computationally more expensive.

6.   CONCLUSION AND QUESTIONS FOR FURTHER RESEARCH

In this paper we introduced a new strategy for Mastermind, called Most-Parts Strategy, which is easy to calculate
and performs best among the five presented easily computed strategies on the standard Mastermind game. In
the range of possible strategies based on partitions generated by guesses, it is an extreme. It only looks at the
"breadth" (size) of a partition. On the other side of the spectrum is Knuth's Worst-Case Strategy which only looks
at the "depth" (maximal element) of a partition. The Expected-Size Strategy and the Entropy Strategy seem to
find a midway between these two extremes. There are probably many more strategies that can be found.

One of the anonymous referees pointed out that the selection of the first question is crucial. The first question
should be $AABC$, just as in Koyoma and Lai's (1993) optimal strategy. It seems that the standard version of
Mastermind is quite limited with respect to these strategies. It might be worthwhile to look at other versions of
the game to be able to tell how well these strategies do in general. However, that was beyond the scope of this
paper. Fortunately, there are still many questions remaining about Mastermind.

7.   REFERENCES

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley Series in Telecommunications. John
Wiley & Sons Inc.

Irving, R. (1978-1979). Towards an Optimum Mastermind Strategy. *Journal of Recreational Mathematics*,
Vol. 11, No. 2, pp. 81–87.

Knuth, D. (1976-1977). The Computer as Master Mind. *Journal of Recreational Mathematics*, Vol. 9, No. 1, pp.
1–6.

Koyoma, K. and Lai, T. (1993). An Optimal Mastermind Strategy. *Journal of Recreational Mathematics*, Vol. 25,
No. 4, pp. 251–256.

Neuwirth, E. (1982). Some Strategies for Mastermind. *Zeitschrift für Operations Research*, Vol. 26, pp. B257–
B278.

Shapiro, E. (1983). Playing Mastermind Logically. *SIGART Newsletter*, Vol. 85, pp. 28–29.

Sterling, L. and Shapiro, E. (1994). *The Art of Prolog: advanced programming techniques*. MIT Press, Cam-
bridge, Massachusetts, second edition.