YET ANOTHER MASTERMIND STRATEGY

Barteld Kooi[1]

Department of Philosophy, University of Groningen, The Netherlands

ABSTRACT

Over the years a lot of easily computable strategies for the game mastermind have been proposed. One of the obvious strategies, guess that code that has the most possible answers, has been lacking. It is discussed in this paper.

1. INTRODUCTION

Mastermind is a two-player zero-sum game of imperfect information. First player I chooses a combination of four pawns drawn from six colors. Player II does not know the choice player I made. Then player II can ask up to eight questions in the form of a combination. If she asks the secret combination, she wins the game, otherwise player I wins the game. Each time player II asks a question, she gets an answer that expresses the accuracy of the question. The answer consists of two number: the number of pawns that are of the right color and in the right place, and the number of pawns that are of the right color, but are not in the right place. For example:

$AABB$ the secret combination
$BBAB$ the question

In this case the answer is: one pawn is in the right place and two pawns have the right color but are not in the right place (I will abbreviate this as (1,2)).

Player II has many winning strategies that guarantee that the secret combination is found within eight questions. There is even a strategy that guarantees that the solution is found within five questions (see below). One can also ask seven questions simultaneously, and deduce the secret from the answers. 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 on the game. At the start of the game eight dollars are available. Player I gets as many dollars as the number of questions player II asks and player I gets the rest. Now the question is which strategy minimizes the expected number of questions required, thus maximizing player II's expected payoff. To be able to calculate the expected number of questions, one needs to make an assumption about player I's strategy. In most papers it is assumed that player I chooses a secret combination at random (by a uniform distribution). I also make that assumption in this paper.

Many papers have been written about mastermind since the game was first sold in the 1970's. One paper seems to be the definitive paper on mastermind strategies by Koyoma and Lai (Koyoma and Lai, 1993). They found the optimal strategy by depth first search on a supercomputer. With this strategy the expected number of questions required by player II is 4.340. In most of the earlier papers strategies are put forward which can be calculated easily. It still seems worthwhile to study these strategies. Although these are not optimal, their computational complexity makes them easily generalizable to other setting and variations of the game (more colors or more positions). In this paper I only look at the standard version of the game.

From the easily computable strategies that have been proposed over the years there is one very simple one that is lacking: ask the question that has most possible answers. Nevertheless, from the easily computable

[1]email: barteld@philos.rug.nl

strategies it has the least expected number of questions: 4.373. This surprisingly simple strategy is discussed in this paper.

In Section 2 some of the well-known mastermind strategies that are easily computed are presented. In Section 3 the new strategy is motivated and presented. In Section 4 the empirical results are presented when these strategies are played. In Section 5 these results are discussed and possible explanations why some strategies are not doing well are also discussed. In Section 6 some conclusions are drawn and questions for further research are indicated.

2. SOME WELL-KNOWN MASTERMIND STRATEGIES

In this section I want to introduce some of the strategies that have been proposed over the years in order to compare them with the stategy that asks the question with the most possible answers.

2.1 A Simple Strategy

The first strategy by Shapiro (Shapiro, 1983) (it is also published in (Sterling and Shapiro, 1994)). His algorithm does is the following: the combinations are somehow ordered (usually alphabetically) and the first combination is asked. The answer is received. The next question is the first one in the ordering that is consistent with the answers given so far. And so on until the combination is cracked. A crucial drawback to this strategy, however, is that it looks at the informativity of questions very marginally. One can only be certain that one does not know the answer already, but that is all.

2.2 Looking One Step Ahead

In mastermind a question partitions the set of possible combinations. This can be seen in the following example. Consider a simplified mastermind game with two pawns and four colors. The set of possible combinations can be represented as follows:

$$\begin{array}{cccc} DA & DB & DC & DD \\ CA & CB & CC & CD \\ BA & BB & BC & BD \\ AA & AB & AC & AD \end{array}$$

The questions AA and DA can be represented by the corresponding answers as:

$$\begin{array}{cccc} 1,0 & 0,0 & 0,0 & 0,0 \\ 1,0 & 0,0 & 0,0 & 0,0 \\ 1,0 & 0,0 & 0,0 & 0,0 \\ 2,0 & 1,0 & 1,0 & 1,0 \end{array} \qquad \begin{array}{cccc} 2,0 & 1,0 & 1,0 & 1,0 \\ 1,0 & 0,0 & 0,0 & 0,1 \\ 1,0 & 0,0 & 0,0 & 0,1 \\ 1,0 & 0,1 & 0,1 & 0,2 \end{array}$$

It is obvious that the second question is more informative than the first, but how can we motivate this intuition? The idea of all the strategies presented below is that the choice for a question is based solely on the partition of the remaining possibilities that a question generates. So in the simplified game above two different kinds of questions can be asked at the start of the game. This is summerized in the following table.

	AA	AB
$(0,0)$	9	4
$(0,1)$	0	4
$(0,2)$	0	1
$(1,0)$	6	6
$(2,0)$	1	1

	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
(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 1: First possible questions. For each question and each answer, the number of combinations that would yield that answer to the question is given. The positive numbers can also be seen as the sizes of the elements of the partition generated by the question.

A number in a cell represents the number of combinations in which the answer in the row is given to the question in the column. For example, in the table above, there are 9 combinations where the answer is $(0, 0)$ when the question is AA. Let us look at the partitions for the standard mastermind game; four pawns and six colors provided in Table 1. The numbers in this table are interpreted in the same way as in the table above. For example, there are 625 combinations where the answer is $(0, 0)$ when the first question is $AAAA$. It seems obvious that question $AAAA$ is not a good question, but what more can be said?

2.3 A Worst Case Strategy

One can look at Table 1 from a worst case perspective. If player II wants to minimize the number of questions required to guess the secret combination, the number of combinations player II considers possible, gives an indication of the number of questions it will take. The worst thing that can happen to player II, is that the answer to a question leaves her with the largest element of the partition.

	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
the size of the largest partition element	625	317	256	276	312

Assuming player II wants to play it safe, she should ask the question $AABB$, because this minimizes the largest partition element. This strategy is presented in (Knuth, 1976-1977).

2.4 An Expected Size Strategy

One could also say that when player II decides which question she should ask, her choice should not be based on the worst case, but on the 'expected case', because she wants to maximize her expected payoff. Therefore, one should look at the expected size of the partition element one ends up with. 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. Let A be the set of possible answers to questions. Let g be a question, then the expected size of the partition

element one ends up with is:

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

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

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

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

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

For the first question the expected sizes are shown in the table below

	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
the expected size of a partition element	511.9	235.9	204.5	185.3	188.2

From this point of view, player II should ask $AABC$ as the first question. This approach is taken in (Irving, 1978-1979)[2].

2.5 An Information Theoretical Strategy

There is a measure that gives an ordering on partitions which is called entropy (see (Cover and Thomas, 1991)). The concept entropy plays an important role in information theory for measuring the amount of information of messages. One can also use it for a mastermind strategy. This strategy can be motivated by the following example. Suppose we have a guessing game. Player I picks a card randomly from a deck of cards. Player II has to determine which card Player I picked using as few yes/no questions as possible. If there are eight cards for example, 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. (It is not exactly the expected number of questions, because one should look at the logarithm as the limit of the expected number of questions, if one can play a number of these games simultaneously.) Suppose we have a partition $V = \{V_1, \ldots, V_m\}$ of a set A. Let p_i be $\frac{\#(V_i)}{\#(A)}$. (This is the probability that an element of V_i is in A. If the probability distribution is not uniform another definition is needed.) Then the expected number of yes/no questions could be represented as

$$\sum_{i=1}^{n} p_i \log(\#(V_i))$$

Trying to minimize this measure is the same as trying to maximize the entropy which is defined as

$$-\sum_{i=1}^{n} p_i \log(p_i)$$

[2]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 questions required for all 1296 combination is 5804, whereas it is 5801 according to my calculations. This can be explained by a minor programming error (the same that I made), but I cannot explain any of his other results. He says his strategy selects the first two questions on the basis of the expected number of remaining possibilities and the rest by exhaustive search. When I look at the second questions to be asked according to his strategy I disagree with him on five questions. In four of those it is simply the case that he does not take the first one out of the list that is available to him. In one case it is simply wrong. His first question is $AABC$. If the reply to this question is (3,0), according to Irving the next question should be $FBAC$. (One immediately wonders why not $DBAC$.) According to my calculations, the expected size of the set of remaining possibilities after this question is 4.7. However, if one asks $ABCC$ the expected size is 3.6, which is quite different. One difference between these two questions is that Irving's question partitions the remaining possibilities in 8 parts, whereas $ABCC$ partitions the set of remaining possibilities in 7 parts. So it might be the case he took the average number of remaining possibilities, instead of the expected size, but I still cannot reproduce his results.

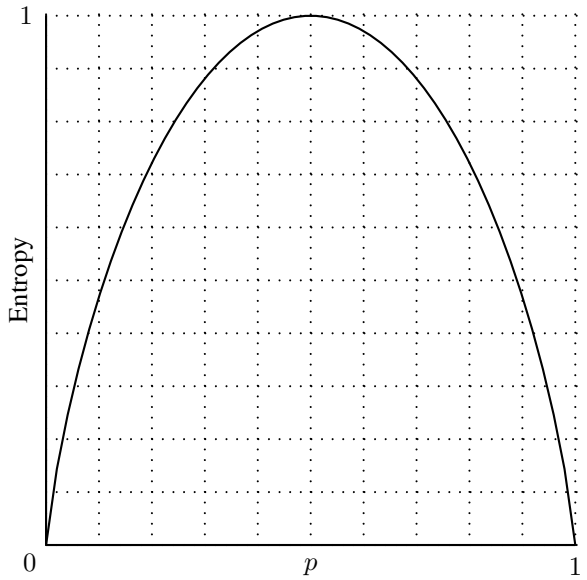

Figure 1: Entropy of a partition with two elements.

since $\log(p_i) = \log\left(\frac{\#(V_i)}{\#(A)}\right) = \log(\#(V_i)) - \#(A)$. In Figure 1 a graph displaying the entropy for partitions with two elements is drawn. 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, for a partition with two elements entropy would select the partition where 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 questions.

	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
entropy	1.498	2.693	2.885	3.044	3.057

From this perspective player II should start by asking $ABCD$. This strategy is one of the strategies studied in (Neuwirth, 1982).

3. A NEW STRATEGY

One can also take another approach to guessing games. Suppose again that player II has to guess which card player I drew randomly from an ordinary deck of cards. Player II wins 1$ if the guess is correct. There are 52 possibilities. Before player II guesses she can ask one yes/no question, which is truthfully answered by player I. 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. Surprisingly all yes/no question are equally good. This can be seen as follows. Suppose the two piles have sizes x and y. The card is in group x with probability $\frac{x}{52}$. The probability of guessing the right card if it is in this group is $\frac{1}{x}$. The card is in group y with probability $\frac{y}{52}$. The probability of guessing the right card if it is in this group is $\frac{1}{y}$. 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 can be generalized. Suppose there is a set A and we have to guess what element of A we are dealing with. We also have to assume that the probability distribution on 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:

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

So in these cases the sizes of the elements of the partition do not matter. The only thing that matters is the size of the partition, i.e. the number of elements of the partition.

This can also be generalized to games with more rounds. Assume that the probability of guessing the element of any set A correctly in a game with r rounds is the number of parts that one can partition A in with r questions, divided by the cardinality of A, where the player's question can depend on the answer to the previous questions. Let us look at a game with $r+1$ rounds. A player can ask $r+1$ questions, and then has to guess which element of A the other player chose. Let the first question be a partition $V = \{V_1 \dots V_n\}$. Let n_i indicate the number of parts V_i can be partitioned in 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 A is:

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

So the probability of guessing the element of set A correctly in a game with $r+1$ rounds equals the number of parts that one can partition A in with $r+1$ questions divided by the cardinality of A. By induction one can conclude that this holds for any r and any set A.

This game is also very much like mastermind. So in mastermind, if one wants to maximize the number of combinations for which one would win in a certain round, then one should maximize the number of parts the set of all combinations is partitioned in, in the previous round. It is still not feasible to calculate this for an interesting number of rounds, such as five, but it can be used as a motivation for a strategy. Let us look at the first question again, then we see:

	$AAAA$	$AAAB$	$AABB$	$AABC$	$ABCD$
partition elements	5	11	13	14	14

So this strategy should start with either $AABC$ or $ABCD$. When one writes a computer-program, however, one has to make a choice. In most of the literature an alphabetical ordering is used and I also used this. So, first the questions that maximize the number of parts are selected. Secondly, from these the consistent questions are selected, if possible. Then alphabetical order is used to select a question. Therefore the first question player II asks when she uses this strategy is $AABC$. This is of course a bit arbitrary, and it seems a pity that something as important as the first question relies on it.

4. EMPIRICAL RESULTS

The first table shows for each strategy for how many combinations the game is won in a particular round of the game. Or put in other words: each strategy produces a game tree, the table shows for each depth of the tree how many leafs (nodes without successors) there are.

Round number	1	2	3	4	5	6	7	8	9
Simple	1	4	25	108	305	602	196	49	6
Worst case	1	6	62	533	694	0	0	0	0
Expected size	1	10	54	645	583	3	0	0	0
Entropy	1	4	71	612	596	12	0	0	0
Most parts	1	12	72	635	569	7	0	0	0

The second table shows the same results, but shows for how many combinations the game has been won before or at the end of a particular round, i.e. the numbers in the table above are added.

Round number	1	2	3	4	5	6	7	8	9
Simple	1	5	30	138	443	1045	1241	1290	1296
Worst case	1	7	69	602	1296	1296	1296	1296	1296
Expected size	1	11	65	710	1293	1296	1296	1296	1296
Entropy	1	5	76	688	1284	1296	1296	1296	1296
Most parts	1	13	85	720	1289	1296	1296	1296	1296

The third table shows how many questions 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 questions needed (the expected length of a path to a leaf). The numbers in the second column are rounded.

	total number of questions	expected number of questions
Simple	7471	5.765
Worst case	5801	4.476
Expected size	5696	4.395
Entropy	5722	4.415
Most parts	5668	4.373

The last four compare quite favourably to the Koyoma and Lai's result of 4.340 (Koyoma and Lai, 1993).

5. EVALUATION

In this section I will try to say something more about the empirical results. It seems quite surprising that the simple strategy performs so badly regarding the maximum number of rounds required and the expected number of rounds required. It does not even guarantee that one wins in eight rounds. It seems that the first question that is asked is not a good choice. This can easily be improved by choosing another combination than $AAAA$ to be the first combination that is asked and let the rest be ordered alphabetically. $AABB$ for example gives the following results:

Round number	1	2	3	4	5	6	7	8	9
Simple strategy starting with $AABB$	1	12	71	253	588	286	78	7	0

which is considerably better. But it still performs badly in comparison to the other strategies. One of the reasons can be explained by the following example. Suppose the are six combinations remaining: $ABAA$, $ABAB$, $ABAC$, $ABDE$, $AEAE$, $ACAE$. Now look at the following table, where for each of these remaining possibilities the answers is shown for asking the question in the column.

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

A consistent question (i.e. a question that is possibly the secret combination) would not be able to distinguish all six combinations, but the question $ABFA$, which is not one of the six remaining combinations, can, as can be seen in the table. In this way the maximum number of questions required and the expected number of questions required can be reduced. In all other strategies except the simple strategy inconsistent questions occur.

One of the other interesting results is that, although strategies often have no theoretic way to distinguish two questions, but only alphabetic ways of distinguishing, the empirical results give a different answer.

Due to a programming error the first tests that I ran had strategies that picked the alphabetically last optimal combination if a unique optimal combination was not in the set of remaining possibilities. These give a slightly different picture.

Round number	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

The simple strategy has been left out of this table, because these considerations do not affect his strategy. These differences are very small. They are greatest in case of Knuth's strategy of minimizing on the maximum size of the partition elements. 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 Knuth's case is because of the following. After the first question 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 question that can be asked in the initial state. But after the first question has been answered there are much more. The following table shows the number of questions that can be asked if the first answer is $(1, 0)$.

question	answer	number of different questions
$AAAA$	$(1, 0)$	12
$AAAB$	$(1, 0)$	53
$AABB$	$(1, 0)$	34
$AABC$	$(1, 0)$	125
$ABCD$	$(1, 0)$	52

So in Knuth's strategy, there are already 34 different kinds of partitions that can be made. His 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 questions that can be asked after the first question, this will be worse after more questions.

The "expected size" strategy is straightforward, and indeed it requires 6 round, 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 results in the best strategy when one looks at the expected number of questions, the only problem is that the theory behind it tells you that the number of rounds really matters, whereas this is ignored in selecting a question. When one looks at the second table in Section 4, one sees that if the number of rounds were only 2,3, or 4, the most parts strategy is better than the other strategies. But in calculating the next question the strategy only looks one step ahead. I found the following looking two

steps ahead.

	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

The numbers in the table represent the number of different answers one could get by asking a question, after the initial question given and the initial answer. So the total number at the bottom is the total number of parts of the partition that results from asking two questions. So if the game consists of three rounds, it is best to start with $AABC$. However looking two steps ahead is computationally more costly.

6. CONCLUSION AND QUESTIONS FOR FURTHER RESEARCH

In this paper I introduced a new strategy for mastermind, which is easy to calculate and does best from all easily computed strategies on the standard mastermind game. In the range of possible strategies based on partitions generated by questions it is an extreme. One only looks at the "breadth" of a partition. On the other side of the spectrum one finds Knuth's worst case strategy, which only looks at the "depth" of a partition. The expected size, and entropy strategies seem to find a mean between these two extremes. There are probably many more means that can be found.

One of the anonymous reviewers pointed out that the selection of the first question is crucial. The first question should be $AABC$, just as in Koyoma and Lai's strategy. It seems that the standard version of mastermind is quite limited if one looks at these strategies, so it would be a good idea to look at other versions of the game, to be able to say how well these strategies do in general. That, however, was beyond the scope of this paper. Luckily there are still many questions remaining about mastermind.

ACKNOWLEDGEMENTS

I thank Johan van Benthem, Marc van Duijn, Wiebe van der Hoek, Erik Krabbe, Gerard Renardel, Rineke Verbrugge and three anonymous reviewers for their comments.

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, Cambridge, Massachusetts, second edition.