NEAR-OPTIMAL STRATEGIES FOR THE GAME OF LOGIK

Albrecht Heeffer[1] *and Harold Heeffer*[2]

Ghent University, Belgium

ABSTRACT

Logik is an interesting variation of the game of Mastermind. For the latter several strategies have been proposed. We present some computational results for existing and new strategies applied to Logik. Our results give some indication on the scalability and applicability of these strategies to similar games.

1. INTRODUCTION

Logik is a two-person zero-sum game with imperfect information which rules are very similar to Mastermind. The major difference is that Logik is played with five pegs of eight colors while Mastermind uses four pegs of six colors. Mastermind has been well analyzed and several strategies guarantee a win in six moves, one even a win in five moves. According to the rules of Mastermind, the game can thus always be won by the player who has to guess the code. Logik is more interesting in this respect because both players set up a secret code and have to guess the opponent's code. It is won by the player who first solves the opponent's code. If both players solve the code within the same numbers of moves then the game is a draw. For Mastermind an optimal strategy has been determined by means of an exhaustive depth-first search leading to an average of 4.340 moves (Koyama and Lai, 1993). For Logik it is much more difficult to determine an optimal strategy. While the game uses only one more peg and two more colors, the number of codes is raised from 6^4 to 8^5. It is therefore important to find a near-optimal strategy which is computational inexpensive and easy to implement. Several of these strategies have been proposed for Mastermind. Their performance for the game of Logik provides us an insight into the scalability of the strategies and their applicability to more generalized versions of guessing games. A new strategy, called maximizing the partitions, was recently proposed in this journal by Kooi (2005). The surprising result was that this counter-intuitive strategy scored very close to the optimal for the game of Mastermind. Our study was motivated to find out its performance for Logik. We introduce some new strategies also implementing the idea of equal distribution.

2. GAME CHARACTERISTICS

Logik uses the German spelling of 'logic' possibly because this game is mostly played in Germany and Central Europe. It is featured on the popular game site brainking.com. We will folow their rules and conventions. Let the five pegs be A, B, C, D, E and the eight colors, b: black, c: cyan, g: grey, r: red, y: yellow, l: blue, p: pink and w: white. After placing a guess markers indicate the number of pegs with correct color and place (black marker) and correct color (white marker). We assigned return codes to the possible combination of markers as shown in Table 1. Remark that the situation with 4 black markers and one white cannot occur.

While the first move can theoretically be any of $8^5 = 32768 = 2^{15}$ combinations, there are actually only seven different cases to consider, e.g.: bbbbb, bbbbc, bbbcc, bbbcg, bbccg, bbcgr and bcgry. Other possibilities are permutations of these codes or variations of the chosen colors. For each of these cases we can determine the number of possible solutions for each return code. For example, if we start with bcgry and the return code is 0

[1]Center for Logic and Philosophy of Science, Ghent University, Belgium (albrecht.heeffer@ugent.be)

[2]Department of Engineering, Ghent University, Belgium (harold.heeffer@ugent.be)

Table 1: : all possible combination of markers after each guess

code	black	white	code	black	white
0	0	0	10	0	4
1	0	1	11	1	3
2	1	0	12	2	2
3	0	2	13	3	1
4	1	1	14	4	0
5	2	0	15	0	5
6	0	3	16	1	4
7	1	2	17	2	3
8	2	1	18	3	2
9	3	0	19	5	0

(no colors correct) then only solutions with the three remaining colors are possible, i.e. 3^5 solutions. If we get return code 13 (3 black, 1 white) for any chosen starting guess the number of remaining solutions is 35. Table 2 shows the number of remaining solutions for different first guesses generated by a program. An entry in the table we call a partition of solutions.

Table 2: : The number of solutions remaining after the first guess for all possible result codes

code	bbbbb	bbbbc	bbbcc	bbbcg	bbccg	bbcgr	bcgry
0	16807	7776	7776	3125	3125	1024	243
1	0	7926	6480	7960	7105	5196	2625
2	12005	7585	6480	4467	3796	2387	1280
3	0	1105	2196	4890	4962	7051	7070
4	0	3912	4548	5436	5504	5432	4880
5	3430	2668	2287	2014	1796	1523	1250
6	0	0	336	796	1450	3095	5610
7	0	508	902	1892	2400	3510	4680
8	0	684	1026	1179	1344	1497	1650
9	490	438	412	399	386	373	360
10	0	0	19	36	161	429	1215
11	0	0	72	204	296	652	1120
12	0	78	117	231	282	396	510
13	0	48	72	84	96	108	120
14	35	35	35	35	35	35	35
15	0	0	0	0	4	12	44
16	0	0	3	6	9	24	45
17	0	0	0	6	8	14	20
18	0	4	6	7	8	9	10
19	1	1	1	1	1	1	1

The number of solutions is a monotonously decreasing function of the depth. With each guess (different from the previous) new information is available which narrows the number of solutions. If no solutions remain, then the last guess was the correct one. This class of games can be implemented as a search without an evaluation function. The search has to minimize the number of solutions. We will now discuss different strategies for implementing such a search.

3. NEAR-OPTIMAL STRATEGIES

As Kooi (2005) already gave a comprehensive overview of the different strategies, we will only summarize the basic characteristics of the known ones, and discuss the new. We will not deal with strategies that under perform in Mastermind such as the simple strategy of Shapiro (1983) or Stirling and Shapiro (1994).

3.1 Worst case strategy (Knuth, 1976)

This is the most intuitive strategy. If one wants to minimize the number of possible solutions, it is evident from Table 2 that bbcgr is the best starting move. The maximum number of solutions for any given return code is 7,051, the lowest value in any of the columns. One can consider this strategy a minimax one in which the starting player selects the code leading to the minimal number of solutions and the opponent selects the return code leading to the partition with the maximum number. In Mastermind this strategy leads to an average length of 4.476 moves, which is not the shortest one, but the only one which always leads to a solution of maximum 5 moves.

3.2 Maximum entropy strategy (Neuwirth, 1982)

This strategy based on information theory was suggested by Neuwirth (1982). The information content of the secret code can be seen as the number of yes/no questions one has to ask to determine the code. Reducing the number of such questions is the same as maximizing its entropy. For Logik, the entropy is at its highest if the number of solutions for every return code (the columns in Table 2) is equal. If p_i is the probability that a solution is within partition i of n, then the entropy can be expressed as:

$$-\sum_{i=1}^{n} p_i \log_2 (p_i)$$

For Mastermind this strategy leads to an average length of 4.415, which is slightly better than the worst case strategy. For Logik, the guess bbcgr for the first move also gives the highest entropy (see Table 3).

3.3 Expected size strategy (Irving, 1978)

This strategy aims to minimize the expected solution size of a guess. This can be defined as the probability of getting the answer corresponding to that guess, multiplied with the size of the partition x_i, e.g:

$s = \sum_{i=1}^{n} p_i x_i$, but as $p_i = \frac{x_i}{\sum_{i=1}^{n} x_i}$ and $\sum_{i=1}^{n} x_i$ is the same for all partitions,

we can implement this strategy by taking the sum of the squares of the sizes:

$$s' = \sum_{i=1}^{n} x_i^2$$

The idea can be put more intuitively as follows. We prefer the first of the two sets of partitions (5, 10, 5) and (10, 10, 0), because this one has the best distribution of solutions. The sum of squares is a measure for this and the first set is preferred as 150 is smaller than 200. This strategy can more aptly be called minimum sum of squares. We also tried some more moderate and more aggressive measures such as $\sum_{i=1}^{n} x_i^{3/2}$ and $\sum_{i=1}^{n} x_i^3$

3.4 Maximum partitions strategy (Kooi, 2005)

Probably the easiest way of quantifying the spread of solutions over the return codes is counting the partitions with $x_i > 0$. This strategy, proposed by Kooi (2005), performs surprisingly well and with an average length of 4.373 approaches the optimal strategy for Mastermind.

3.5 Lowest standard deviation strategy

We looked at other ways for selecting the best distribution and found in the standard deviation an indicator of the statistical dispersion of solutions over the return codes. For x_i being the number of solutions for n return codes, the guess with the lowest standard deviation ρ gives the best distribution:

$$\rho = \sqrt{\frac{n \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} (x_i)^2}{n(n-1)}}$$

For the first move, this leads to the choice for bbcgr, the same as previous strategies.

3.6 Random and next choice

Swaszek (1999-2000) found that a random guess from the set of remaining candidate code sequences in Mastermind leads to the surprisingly short average game length of 4.638 and picking the next choice of codes consistent with previous return codes achieves a length of 4.758. We also investigated these "strategies" for Logik. In order to compare them to other strategies we started from the same first guess bbcgr.

Table 3: : The best first guess (in bold) according to the different strategies

strategy	**bbbbb**	**bbbbc**	**bbbcc**	**bbbcg**	**bbccg**	**bbcgr**	**bcgry**
worst case	16807	7776	7776	7960	7105	**7051**	7070
exp. size	13385	6268	5412	4775	4358	**4283**	4369
entropy	1.467	2.642	2.877	3.063	3.180	**3.238**	3.232
partitions	6	14	18	19	**20**	**20**	**20**
stdev	4501	2826	2551	2326	2166	**2136**	2170

4. RESULTS

We implemented a program that generates the full game tree for every given strategy. Thanks to the specific conditions of Logik, all solutions can be represented by consecutive 15-bit numbers which allows for a very efficient C-language implementation with bit shifts and logical operators. Still, one run takes about one full day. Our results are summarized in Table 4 and are rather different from the case of Mastermind.

Table 4: : Number of solutions against number of guesses and average solution length

strategy	1	2	3	4	5	6	7	8	9	**avg**
worst case	1	19	230	1971	10233	16326	3848	139	1	5.670
exp. size	1	18	222	2069	11274	16207	2871	106	0	5.601
entropy	1	18	222	2108	11559	16082	2709	69	0	5.583
partitions	1	18	222	2031	10245	15530	4443	271	7	5.693
stdev	1	18	221	2001	9922	14496	5421	657	31	5.761
random	1	8	226	1865	8829	14312	6533	947	47	5.856

Where maximum partitions scores very well for Mastermind, we now see that most of the other strategies perform better. Two strategies, expected size and entropy, guarantee a solution in eight or less guesses. The best strategy for Logik is maximum entropy with an average solution length of 5.583 followed by the sum of squares (expected size). Raising or lowering the exponent affects but does not improve the original result of 5.601. The sum of cubes leads to a length of 5.653 and the power of $3/2$ to 5.643. The important lesson to be learned form this empirical study is that the performance game strategies such as the ones discussed depend very much specific quantitative parameters of the game. A general assessment of strategies requires a more extensive study of different parameters and generic cases.

5. FURTHER WORK

So far, there have been no attempts at an exhaustive search for an optimal strategy of Logik or other generalizations of Mastermind. A measure of the optimal average length would allow us to better assess near-optimal strategies. In any case, the range between the optimal average length (¡= 5.583) and the length for random choice (5.856) will be small one. A strategy to select a suitable secret code, to maximize the number of moves for the opponent, seems to be bear little fruit. The cryptologist Michael Wiener (1995) found that for Mastermind the best secret code can be chosen at random from all combinations that do not have all the same color. Static Mastermind, as it was coined by Alex Bogomolny and Don Greenwell provides an interesting area for further research. This variation addresses the question how many guesses, made all at once, are needed to find the secret code. For Mastermind this appears to be six. Goddard (2003, 2004) looked for the minimum number of codes for the general case of n pegs and k colors. However, the case of Logik has not yet been reached. While traditional board games such as chess and shogi could be approached as a search for positions in which the number of moves of the opponent is zero, i.e. checkmate, the strategies described here cannot be applied because the number of moves does not decrease monotonously with the search depth. However, there may be some applications for mate search in board game situations and for several card games.

6. REFERENCES

Goddard, W. (2003) ”Static Mastermind”, J. Combin. Math. Combin. Comput., 47, pp. 225-236.

Goddard, W. (2004) ”Mastermind revisited”, J. Combin. Math. Combin. Comput., 51, pp. 215-220.

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

Knuth, Donald E. (1976-77) “The Computer as Master Mind”, *Journal of Recreational Mathematics*, 9, pp. 1-6.

Koyama, K. and Tony W. Lai, (1993) “An Optimal Mastermind Strategy”, *Journal of Recreational Mathematics*, 25, pp. 251-6.

Kooi, B (2005) “Yet another mastermind strategy”, *ICGA Journal*, 28 (1), pp. 13-20.

Neuwirth, E. (1982) “Some Strategies for Mastermind”, *Zeitschrift für Operations Research*, Vol. 26, pp. 257-278.

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, Ma.

Swaszek, P. F. (1999-2000) “The Mastermind Novice”, *Journal of Recreational Mathematics*, 30, pp. 193-198.

Wiener, M. (1995) Usenet, sci.math, 29 Nov 1995.