

Proposal for Master Minds

Xinhao Su
xs2413

David Xu
dx2199

December 3, 2021

1 Introduction

Master Mind is a code breaking game that was invented in 1970 by Mordecai Meirowitz. It resembles the earlier paper and pencil game called Bulls and Cows that may date back a century. The gameplay of Master Mind is straightforward:

1. The codemaker picks 4 pegs (which can each be one of 6 different colors), and places them in order as the code. (Note that the code can have repeated colors, e.g. red red blue blue)
2. The codebreaker needs to guess both the color and the order of the code within 8 turns.
3. At each turn, the codemaker provides feedback by placing 0 to 4 pegs in response to a guess. A black peg indicates that one of the pegs in the guess is correct in both color and order. A white peg indicates that one of the pegs in the guess is correct in color but not order.
4. The game is terminated when the codebreaker's guess is correct or 8 incorrect guesses have been made.



Since we will use parallelism to find the solution, our project will be called *Master Minds*.

2 Algorithm

The Master Mind game can be rephrased as The Mastermind Problem: Given a set of guesses and the feedback for each guess (number of colored and white pegs), is there at least one code that generates those exact feedback responses? It is interesting to note that this problem is in effect a multi-dimensional search problem. The Mastermind Problem has been proven to be NP-Complete¹.

Our project will use Donald Knuth's Five-Guess Algorithm² to break the code:

1. Generate the set of possible codes, denoted as S . E.g., $S = \{6^4 = 1296 \text{ codes}\}$
2. Start with an initial guess (initial guess can be hardcoded or randomly selected from S)
3. Play the guess and get response from codemaker
4. If the feedback is all black, we have found the code.
5. Otherwise, remove all the codes that would not produce the most recent response from S . For example, if our guess had two blues and the response is only one white peg, then we know the true code cannot have two blues.
6. For the next guess, select a code (not necessarily in S) which minimizes the maximum possible number of remaining codes in S across all possible true codes.
7. Repeat from Step 3.

3 Testing Plan

Since the Mastermind game has many variations, we intend to test our project on all or some of the following variations:

1. Mastermind - 6 colors, guessing 4 pegs
2. New Mastermind - 8 colors, guessing 4 pegs
3. Word Mastermind - 26 letter, guessing 4 letters
4. Grand Mastermind - 5 colors with 5 shapes, guessing 4 pegs
5. Crazy Mastermind - 12 colors, guessing 4 pegs

Our project could also be used to analyze which codewords minimize the maximum or average number of guesses required based on which initial guess is used for each of these variations.

4 Opportunities for Parallelization

1. The execution of the algorithm with different initial guesses is embarrassingly parallelizable.
2. In the process of selecting the next guess, the calculation of the worst-case (maximum possible) number of remaining codes for any given guess can be parallelized.
3. If we desire to play multiple variations of the game at the same time, such gameplay can also be parallelized.

¹Stuckman, J., & Zhang, G. Q. (2005). Mastermind is NP-complete. *arXiv preprint cs/0512049*.

²Knuth, D. E. (1976). The computer as master mind. *Journal of Recreational Mathematics*, 9(1), 1-6.