Artificial Intelligence: CS 209 / 214
Project Proposal – Group 6

Members:

Amogh R (CS22BT005)
R Amogh (CS22BT046)
Rahul Anand (CS22BT069)
Krutay Upadhyay (CS22BT029)
Subhash Chandra P (CS22BT071)
Siddharth Singh (CS22BT055)

Problem Statement (Mastermind) -
Introduction:

Mastermind is a code-breaking game for two players. It resembles an earlier pencil and paper game called Bulls and Cows that dates back over a century.

The game is played using:

- a *decoding board*, with a *shield* at one end covering a row of four large holes, and twelve (or ten, or eight, or six) additional rows containing four large holes next to a set of four small holes;
- *code pegs* of six different colors with round heads, which will be placed in the large holes on the board; and
- *key pegs*, some colored (green in the image, though often black) and some white, which are flat-headed and smaller than the code pegs; they will be placed in the small holes on the board.

The game is played as follows:

Player 1 (the codemaker) chooses a pattern of four code pegs. The codemaker places the chosen pattern in the four holes covered by the shield, visible to the codemaker but not to Player 2 (the codebreaker).

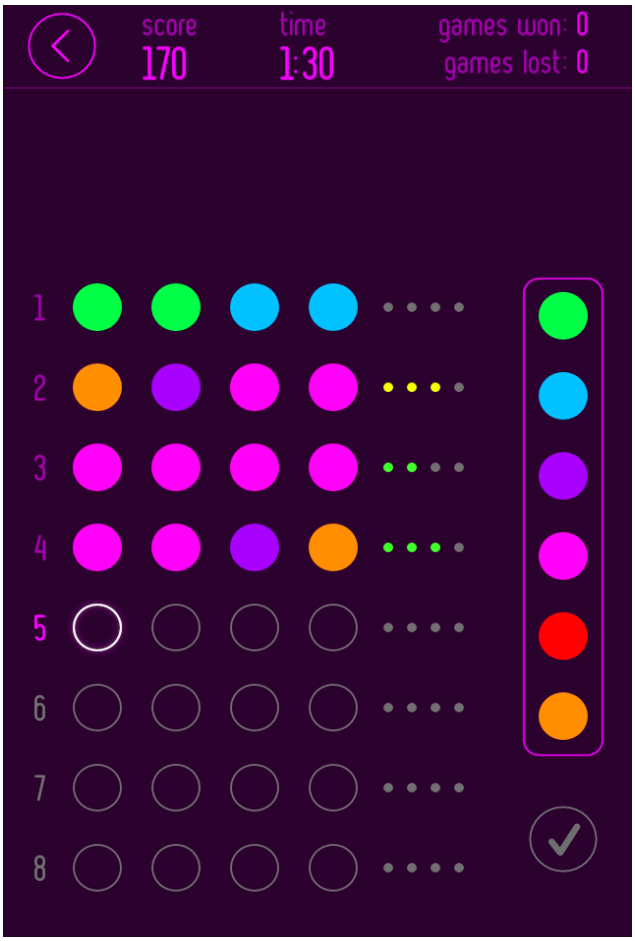
The codebreaker tries to guess the pattern, in both order and color, within a specified number of turns. Each guess is made by placing a row of code pegs on the decoding board. Once placed, the codemaker provides feedback by placing from zero to four key pegs in the small holes of the row with the guess. A (green) colored key peg is placed for each code peg from the guess which is correct in both color and position. A (yellow) non - coloured key peg indicates the existence of a correct color code peg placed in the wrong position.

An electronic version of the game is

shown here.

Variations of the game exist, with some allowing duplicate colours, blank holes in

the code, as well as varying difficulties with limits on the number of guesses, different number of colours and holes,
etc.



==========

Figure: An electronic version of Mastermind

==========

Complexity and Satisfiability:

In November 2004, Michiel de Bondt proved that solving a *Mastermind* board is an NP-Complete problem when played with *n* pegs per row and two colors.

The *Mastermind satisfiability problem* is a decision problem that asks, "Given a set of guesses and the number of colored and white key pegs scored for each guess, is there at least one secret pattern that generates those exact scores?" (If not, then the codemaker must have incorrectly scored at least one guess.) In December 2005, Jeff Stuckman and Guo-Qiang Zhang showed in an arXiv article that the *Mastermind satisfiability problem* is NP-complete.

Previous Research -

In 1977, Donald Knuth demonstrated that the codebreaker can solve the pattern in five moves or fewer, using an algorithm that progressively reduces the number of possible patterns.

Subsequent mathematicians have been finding various algorithms that reduce the average number of turns needed to solve the pattern: in 1993, Kenji Koyama and Tony W. Lai performed an exhaustive depth-first search showing that the optimal method for solving a random code could achieve an average of 5,625/1,296 = 4.3403 turns to solve, with a worst-case scenario of six turns.

Proposal -

The general problem of Mastermind can be modeled as a search problem with a high branching factor of the state space tree. Our project aims to improve upon previous algorithms by using a heuristic based approach to reduce computation.

An simplified version of the game will be implemented first, with gradually increasing complexity (features) with time. Monitoring performance of algorithms used for each stage and modifying them accordingly becomes easier*.

We plan to approach the problem using evolutionary algorithms**, and both improve upon such pre-existing algorithms, and also try to create new strategies for the game.

The initial problem we model would be limited to distict colours being used for a code, which greatly reduces the branching factor and the size of the state space tree from growing exponentially to an order of factorial (for a relatively low number of colours).

, *** Suggestions/Feedback are appreciated

References -

1. Knuth, Donald (1976–1977). "The Computer as Master Mind" (PDF). J. Recr. Math. (9): 1–6.

2. Koyama, Kenji; Lai, Tony (1993). "An Optimal Mastermind Strategy". Journal of Recreational Mathematics (25): 230–256.

3. Berghman, Lotte (2007–2008). "Efficient solutions for Mastermind using genetic algorithms" (PDF). K.U.Leuven (1): 1–15.

4. Merelo J.J.; Mora A.M.; Cotta C.; Fernández-Leiva A.J. (2013). "Finding an Evolutionary Solution to the Game of Mastermind with Good Scaling Behavior". In Nicosia, G.; Pardalos, P. (eds.). Learning and Intelligent Optimization. Lecture Notes in Computer Science. Vol.7997. Springer. pp. 288–293

5. De Bondt, Michiel C. (November 2004), NP-completeness of Master Mind and Minesweeper, Radboud University Nijmegen

6. Merelo-Guervós, J.J., Runarsson, T.P. (2010). Finding Better Solutions to the Mastermind Puzzle Using Evolutionary Algorithms. In: Di Chio, C., et al. Applications of Evolutionary Computation. EvoApplications 2010. Lecture Notes in Computer Science, vol 6024. Springer, Berlin, Heidelberg.

7. Maestro-Montojo, J., Merelo, J.J., Salcedo-Sanz, S. (2013). Comparing Evolutionary Algorithms to Solve the Game of MasterMind. In: Esparcia-Alcázar, A.I. (eds) Applications of Evolutionary Computation. EvoApplications 2013. Lecture Notes in Computer Science, vol 7835. Springer, Berlin, Heidelberg.

8. Runarsson, T.P., Merelo-Guervós, J.J. (2010). Adapting Heuristic Mastermind Strategies to Evolutionary Algorithms. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol 284. Springer, Berlin, Heidelberg.