**Possible strategies:**

With four pegs and six colours, there are 55 = 3125 different patterns (allowing for `duplicate colours).

**Five Guess algorithm:**

In 1977, Donald Knuth demonstrated that this algorithm can solve the pattern in five moves or less. The algorithm progressively reduced the number of possible patterns. It works as follows:

1. Create the set S of 1296 possible codes (1111, 1112 ... 6665, 6666)
2. Start with initial guess 1122 (Knuth gives examples showing that other first guesses such as 1123, 1234 do not win in five tries on every code)
3. Play the guess to get a response of coloured and white pegs.
4. If the response is four coloured pegs, the game is won, the algorithm terminates.
5. Otherwise, remove from S any code that would not give the same response if it (the guess) were the code.
6. Apply minimax technique to find a next guess as follows: For each possible guess, that is, any unused code of the 1296 not just those in S, calculate how many possibilities in S would be eliminated for each possible coloured/white peg score. The score of a guess is the minimum number of possibilities it might eliminate from S. A single pass through S for each unused code of the 1296 will provide a hit count for each coloured/white peg score found; the coloured/white peg score with the highest hit count will eliminate the fewest possibilities; calculate the score of a guess by using "minimum eliminated" = "count of elements in S" - (minus) "highest hit count". From the set of guesses with the maximum score, select one as the next guess, choosing a member of S whenever possible. (Knuth follows the convention of choosing the guess with the least numeric value e.g. 2345 is lower than 3456. Knuth also gives an example showing that in some cases no member of S will be among the highest scoring guesses and thus the guess cannot win on the next turn, yet will be necessary to assure a win in five.)
7. Repeat from step 3.

Other, later 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](https://en.wikipedia.org/w/index.php?title=Tony_W._Lai&action=edit&redlink=1) found a method that required an average of 5625/1296 = 4.340 turns to solve, with a worst-case scenario of six turns. The [minimax](https://en.wikipedia.org/wiki/Minimax) value in the sense of [game theory](https://en.wikipedia.org/wiki/Game_theory) is 5600/1296 = 4.321.

**Genetic algorithm**

A new algorithm with an embedded [genetic](https://en.wikipedia.org/wiki/Genetic_algorithm) algorithm, where a large set of eligible codes is collected throughout the different generations. The quality of each of these codes is determined based on a comparison with a selection of elements of the eligible set. This algorithm is based on a heuristic that assigns a score to each eligible combination based on its probability of actually being the hidden combination. Since this combination is not known, the score is based on characteristics of the set of eligible solutions or the sample of them found by the evolutionary algorithm.

The algorithm works as follows:

1. Set i = 1
2. Play fixed initial guess G1
3. Get the response X1 and Y1
4. Repeat while Xi ≠ P:
   1. Increment i
   2. Set Ei = [∅](https://en.wikipedia.org/wiki/Empty_set) and h = 1
   3. Initialize population
   4. Repeat while h ≤ maxgen and |Ei| ≤ maxsize:
      1. Generate new population using crossover, mutation, inversion and permutation
      2. Calculate fitness
      3. Add eligible combinations to Ei
      4. Increment h
   5. Play guess Gi which belongs to Ei
   6. Get response Xi and Yi

**Complexity and the satisfiability problem**:

In November 2004, [Michiel de Bondt](https://en.wikipedia.org/w/index.php?title=Michiel_de_Bondt&action=edit&redlink=1" \o "Michiel de Bondt (page does not exist)) proved that solving a Mastermind board is an [NP-complete](https://en.wikipedia.org/wiki/NP-complete) problem when played with n pegs per row and two colours, by showing how to represent any [one-in-three 3SAT](https://en.wikipedia.org/wiki/One-in-three_3SAT) problem in it. He also showed the same for Consistent Mastermind (playing the game so that every guess is a candidate for the secret code that is consistent with the hints in the previous guesses).

The Mastermind satisfiability problem is a [decision problem](https://en.wikipedia.org/wiki/Decision_problem) that asks, "Given a set of guesses and the number of coloured and white pegs scored for each guess, is there at least one secret pattern that generates those exact scores?" (If not, then the code maker must have incorrectly scored at least one guess.) In December 2005, [Jeff Stuckman](https://en.wikipedia.org/w/index.php?title=Jeff_Stuckman&action=edit&redlink=1) and [Guo-Qiang Zhang](https://en.wikipedia.org/w/index.php?title=Guo-Qiang_Zhang&action=edit&redlink=1) showed in an [arXiv](https://en.wikipedia.org/wiki/ArXiv) article that the Mastermind satisfiability problem is NP-complete.

Sources:

<http://arxiv.org/abs/1207.1315>

<http://www.dcc.fc.up.pt/~sssousa/RM09101.pdf>

<https://lirias.kuleuven.be/bitstream/123456789/164803/1/kbi_0806.pdf>