Solving Mastermind Using Genetic Algorithms

Tom Kalisker and Doug Camens

Software Engineering Masters Program
Engineering and Information Science Division
Penn State Great Valley School of Graduate Professional Studies
30 East Swedesford Road
Malvern, PA 19355-1443
{trk139,dmc123@psu.edu}@psu.edu

Abstract. The MasterMind game involves decoding a secret code. The classic game is a code of six possible colors in four slots. The game has been analyzed and optimal strategies have been posed by computer scientists and mathematicians. In this paper we will survey previous work done on solving MasterMind, including several approaches using Genetic Algorithms. We will also analyze the solution sets and compare our results using a novel scoring system inside a GA against previous work using Genetic and Heuristic algorithms. Our GA is performing closer to optimal then previously published work. The GA we present is a Steady State GA using Fitness Proportional Reproduction (FPR), where the fitness function incorporates a simple heuristic algorithm. We also present a scoring method that is simpler then those used by other researchers. In larger games such as 10 colors and 8 slots our GA clearly outperform the heuristic algorithm. In fact if one wishes to tradeoff a higher average number of guesses to a faster running time, extremely large games such as 12 x10 can be solved in a reasonable time (i.e. minutes) of run time.

MasterMind is a game where a secret code is discovered through decoding successive guesses with hints as to the quality of each guess.

There have been many papers published about strategies to solve MasterMind, several papers including one by Knuth are summarized here.

Other researchers have looked at rule based or heuristic approaches to solving 6x4 MasterMind. The best to date is that by Radu Rosu.

The solutions using GAs by Bento et al and Team Geneura are reviewed.

The score for any size game of MasterMind can be given by the following formula:

Let N = total number of pegs awarded i.e.(B+W)
B = number of black pegs
W= number of white pegs

$$\text{Score} = (2 * \text{B} + \text{W}) + \sum_{i=1}^{N-1} i$$

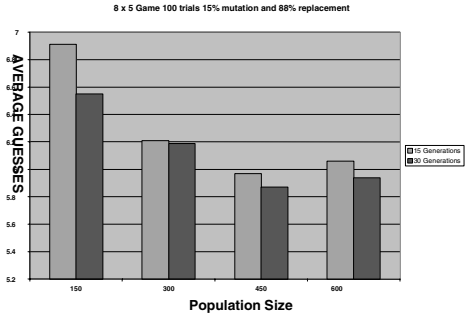
A genome represents a potential guess in the game of MasterMind. It is encoded using an array of integer genes. The alleles possible for each gene are the set of integers from 0 to c-1 where c is the number of colors in the game. The GA consists of a custom fitness function based on the three components listed below.

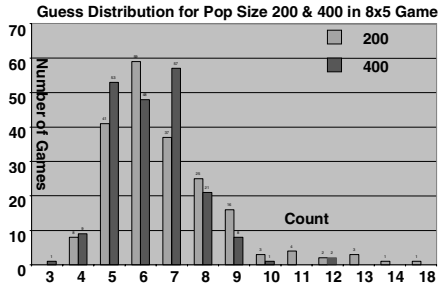
E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1590–1591, 2003.

1. The heuristic algorithm introduced by Radu Rosu.
2. A scoring system based on the number of black and white pegs awarded in previous guesses of the game.
3. Any individual that has already been played in a previous guess automatically gets a fitness of zero. We had hoped that this alone would eliminate the play of multiple guesses.

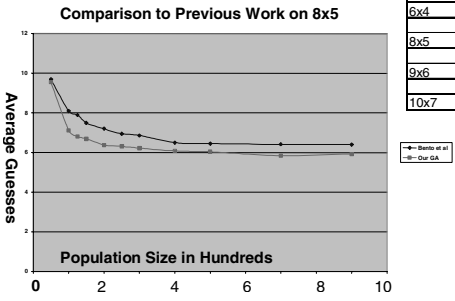
To calculate fitness for an individual, it is evaluated against all previous guesses as if the previous guess is a secret code and the individual is a guess to that code. Simple single-point crossover was used exclusively Mutation is done by randomly switching a gene (i.e. peg) to a different color chosen at random from the colors available in the gene color pool.

We ran our GA with the same size populations as those reported by Bento et al. and show the following results for averages of 500 trials.





Colors x Pegs	Population	Average # Guesses	Avg. Time/Game(SECS)
6x4	120	4.75	0.26
8x5	330	6.39	1.01
9x6	530	7.27	2.31
10x7	1000	8.62	6.72



Acknowledgments. The authors would like to thank Dr. Walter Cedeño and Penn State Great Valley for providing a course in Genetic Algorithms.