Using Evolutionary Algorithms to Solve Sudoku Problems

640034510

October 27, 2017

1 Introduction

The Sudoku puzzle is a combinatorial number placement where the player must fill a partially completed nine by nice grid with the numbers one to nine. These numbers must be placed in a way such that each row, column, and three by three sub grid contain a unique set. Due to this constraint on the placement of numbers, Sudoku proves to be a NP-complete problem which makes it impracticable to solve using a direct method. Instead a heuristic search method must be applied to the problem, of which the genetic algorithm metaheuristic is highlighted in this report. Inspired by the biological process of Meiosis, genetic algorithms encode potential solutions as population of chromosomes which undergo a series of genetic operators to produce offspring. Over successive generations of offspring, the average fitness of the population should increase until the final solution is found.

2 Design

2.1 Solution Space and Representation

The solution space for an empty nine by nine Sudoku grid is vast, with an approximate $6.67 * 10^{21}$ possible grids. As the number of clues given in a Sudoku puzzle increases then the possible solution space decreases but still remains large. To further reduce the size of the solution space, it was decided that all solutions must have valid rows. That is, all rows of an individual must contain some permutation of the set of numbers one to nine. All subsequent operators on a population must therefore conserve this rule to ensure that the search space remains reduced and errant solutions do not enter the population.

Solutions are represented as a one dimensional list of integers, the length of which is set to the number of spaces in the Sudoku grid. This representation was chosen as it makes representing rows, columns, and grids within the solution trivial. Additionally, this representation allows for genetic operators, described below, to be applied to solution chromosomes in an analogous way to that of their biological counterparts.

2.2 Fitness Function

The final success state of a Sudoku puzzle is one where the contents of each row, column, and box are a permutation of the numbers of the numbers one to nine. Therefore the fitness function first calculates the sum of the number of unique digits in each column and box. To normalize this number in the range of zero and one the sum is divided by the total possible number of unique digits in each row and column. In the case of a nine by nine Sudoku grid, the maximum number of unique digits in the rows and columns is 162. Rows have been excluded from this metric as all solutions already have valid rows which present the maximum score. By only including rows and columns, it increases the change in the fitness value for changes to rows and columns which move the solution to the optimum problem. Say something about zero being the worst and one being the best.

2.3 Crossover Function

2.4 Mutation Function

2.5 Population Initialisation

2.6 Selection and Replacement

2.7 Termination Criteria

3 Experiments

4 Questions