

Assignment on Genetic Algorithm

Lab Group : A2

Full Marks:20

ATTENTION

Read the documentation/specification provided thoroughly before you have started doing your assignment. Everything we want have been stated here properly. :)

1 Introduction

You will write a genetic algorithm (GA) to solve a Queen-Bishop placement problem in a chessboard.

2 Problem Statement

You will have a 8×8 chessboard. You will be given 4 pieces of Queens and 4 pieces of bishops. You have to place them in such a way that those pieces do not attack each other. Also, you must ensure that there will be only one piece per column.

3 Moves

3.1 Queen

The queen can be moved any number of unoccupied squares in a straight line vertically, horizontally, or diagonally, thus combining the moves of the bishop and rook (see Figure 2). The queen captures by occupying the square on which an enemy piece sits.

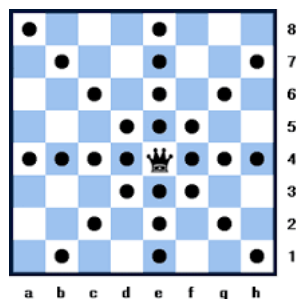


Figure 1: Queen's moves.

4 Moves

4.1 Bishop

The bishop has no restrictions in distance for each move, but is limited to diagonal movement. Bishops, like all other pieces except the knight, cannot jump over other pieces. A bishop captures by occupying the square on which an enemy piece sits.

The bishops may be differentiated according to which wing they begin on, i.e. the king's bishop and queen's bishop. As a consequence of its diagonal movement, each bishop always remains on either the white or black squares, and so it is also common to refer to them as light-squared or dark-squared bishops.



Figure 2: Bishop's moves.

5 Some Definitions

5.1 genome

The list of a population member's genes. Genes are 1s and 0s. You can represent this as a list, e.g., (1 0 0 1 1 0 1 1) In some problems, they are represented as natural numbers (10,1,2,3 etc.) also.

How to apply it in this problem

Your genome will consist of two halves. First half will have 8 integers, either 0 or 1. 0 will denote a queen and 1 will denote a bishop. For example, {1, 0, 0, 0, 0, 1, 1, 1}. It means that first column has a bishop, next four have queens, then next three have bishop. 8 integers ranging from 1 to 8. For example, {1, 5, 3, 6, 2, 7, 8, 4}. It denotes which rows are occupied by pieces in each column. Now, during your crossover, you must crossover the 2nd half, and then adjust the first half accordingly.

5.2 phenome

The expression of an individual's genome in the "world." In this case, the phenome is the form of what's being evolved, or a description of the behavior of the individual, and it comes directly from the genome. The board placement's genome would be a list of the placements of queens, decoded from the genome.

How to apply it in this problem

If you denote a queen by **Q** and a bishop by **R**, then you can build the phenotype like this, using their positions stated in a chessboard: (*Qa1, Bb5, Bc3, Bd6, Qe2, Qf7, Bg8, Bh4*). You have to calculate fitness functions from them. There is no strict rule that your phenotype has to follow this format. But whatever the format is, it should denote the positions of queens and bishops in the board, so that you can be able to calculate the fitness function from it and also encode it back to the genome.

5.3 fitness

How good a genome (or phenome) is. Genomes are sometimes scored directly, but often they are converted into a phenome and the phenome is given a score.

How to apply it in this problem

You can find the number of attacking pairs as the "unfit"-ness function. There are $8C2 = 28$ possible pairs. The ideal placement has no attacking pair in it. So you can calculate fitness function as $(28 - \text{number of attacking pairs})$. You can design your own fitness function too. But remember, the ideal placement will always have the highest fitness and the more ideal a placement is near to, the more fitness it will have. For example, if there are two attacking pairs, fitness will be $28 - 2 = 26$.

Another easy way to build a fitness function is to just invert the fitness score (therefore, a less score will become a great score and vice versa. For example, a fitness score 10 will become $1/10 = 0.1$ and a fitness score 20 will become $1/20 = 0.05$)

5.4 individual

Your GA will operate on individuals. An individual will be a list of the fitness, genome and corresponding phenome.

5.5 population

A group of individuals. The GA operates on these, producing their phenomes from genomes, evaluating each one's fitness, then selecting mates and producing new individuals for the next generation. This will be a list of individuals.

6 GA Pseudo-code to follow

Algorithm 1 A Genetic Algorithm Pseudo-Code	
1:	Choose an initial random population of individuals
2:	Evaluate the fitness of the individuals
3:	repeat
4:	Select the <i>best</i> individuals to be used by the genetic operators
5:	Generate new individuals using crossover and mutation
6:	Evaluate the fitness of the new individuals
7:	Replace the <i>worst</i> individuals of the population by the best new individuals
8:	until some stop criteria

Figure 3: GA Pseudo-code

6.1 comments on the steps

Line 1

Population length, N will be taken from the input file. Input file's first line will contain them separated by spaces. So, do not use static sized arrays for populations. Start with N random permutations of numbers $1 - 8$.

Line 2

Fitness function evaluation is up to you. See what is stated in above sections regarding fitness/evaluation functions.

Line 3

Selecting the best individuals is done in many ways. Each group will have to implement only one strategy based on their group number.

$Group_Number \% 3 == 0$ have to implement **K-tournament Selection with $K = 5$** .

$Group_Number \% 3 == 1$ have to implement **Roulette Wheel Selection**.

$Group_Number \% 3 == 2$ have to implement **Stochastic Universal sampling**.

All groups can read about these strategies from https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm

Line 5

Crossover rate and mutation will be provided by the input file. Input file's second line will contain two double/float values separated by whitespace, denoting crossover rate and mutation rate.

Crossover *Group_Number%3 == 0* have to implement **Swap mutation** .

Group_Number%3 == 1 have to implement **Scramble mutation** .

Group_Number%3 == 2 have to implement **inversion mutation** .

Each group will apply **Davis Order Crossover (OX1)**. All groups can read about these strategies from https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm and https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_crossover.htm.

Line 7

Implement fitness based selection and replacement in this way : (K old individuals $\Rightarrow K/2$ new children \Rightarrow Replace $K/2$ least fit old individuals with $K/2$ new children.

Line 8

Stopping criteria is the iteration number (also known as generations). Input file's third line will contain the iteration number I . You need to run the process stated from step 4 to 7 for I times.

7 Input file

Scan the population length, N , crossover and mutation rates, number of iterations from input text file. Store them globally.

8 Output

After running the algorithm, print all the existing individuals(solutions) ranked by their fitness values in descending order in console (or you can redirect them to dump the output to a output file).

9 Resource

- Use **THIS** link to get a sample code example and explanation .
- Tutorialspoint Link
- An example of genetic algorithm.

10 Guidelines for submissions

- You can use any of these languages (C/C++, JAVA, Python). If you want to pick something else, you must confirm it to me through your CR.
- Each group will consist (2-3) members. So, form groups today and confirm it to me through your CR.
- **Do not copy. Do not even try.**
- Viva will take place, so take preparation both theoretically and practically.