Name- Yash Gandhi BE IT Batch A 2017140014

Task 1: Find out a research paper based on Genetic Algorithm.

Aim:
To apply genetic algorithm for a given problem

Problem Statement: An Adaptive Genetic Algorithm for Solving NQueens Problem

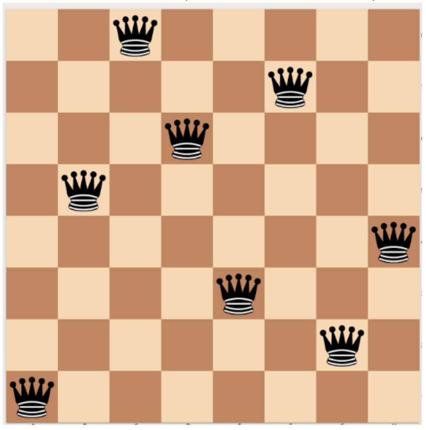
The goal of N Queens Problem is to suitably place N number of Queens on an N x N chessboard in a way that there is no conflict between them due to the arrangement. In chess, a queen can move as far as she pleases, horizontally, vertically, or diagonally. Hence there should be no intersection between them vertically or horizontally or diagonally.
N Queens Problem is an optimization problem attributed to the class of NP-Complete Problems. Heuristic approaches are required to solve N Queens Problem in real time with optimal solutions. Genetic algorithm is one such powerful heuristic method which is capable of efficiently solving the problem.

1. **Description of the component involved in the GA:**
    a. **Type of encoding scheme used in the chromosome representation**

Population Initialization:

Step 1:
Initialize a random population of chromosome of length 1000.
Every chromosome is represented as a vector of length N, which is a random permutation of (1, 2, 3… N).
N-tuple (c1, c2, c3… cN), where ci represents the position of the queen to be in ith column and cth row.

Example:
The chromosome representation for the following instance of queens on the chessboard is (1,5,8,6,3,7,2,4)



b. Type of selection, mutation and crossover schemes used in the paper

Step 2 : Fitness Function:

The fitness function is designed is such a way such that for k number of queens on the same diagonal line will get k-1 points.
The points achieved are summed up to get a fitness value for that chromosome.
In case there is no collision along the diagonal then the fitness value of that chromosome is 0.

Let's take the N x N chess board arrangement (c1, c2, c3… cN).
We'll first check whether more than one Queen is on the same (↗) directed diagonals. This happens if,
For $i \in \{1,2 \ldots N\}$ and $j \in \{1, 2 \ldots N\}$ and $i \neq j$;
$(ci - i) == (cj - j)$
Similarly, for our next checking whether more than one Queen is on the same (↖) directed diagonals, this happens
if,
For $i \in \{1,2 \ldots N\}$ and $j \in \{1, 2 \ldots N\}$ and $i \neq j$;
$(ci + i) == (cj + j)$
Thus we amass the fitness value if any of the above situations occurs.

Crossover Function:
As we are considering each of the chromosomes actually a random permutation of (1, 2, 3… N), so it was needed to design a permutation crossover. Here we are using the order 1 crossover.

Order 1 Crossover is a simple permutation crossover in which 2 random points are selected from parent-1 and the different genes between these points are carried over to the child and the other remaining genes from parent-2 which are absent in child are carried and placed in the child in the sequential order which they appear in parent-2.

Example:

Parent1: 5 2 **3 1 6 4** 8 7 (Select the genes randomly)
Parent2: 1 **8** 6 4 **7 5** 3 **2** (Select the genes not present in Parent1)
Child: 8 7 3 1 6 4 5 2 (Copy Parent1 genes as it is and the rest
In order of Parent 2)

The Crossover Function generates two children from the parent chromosome at the same time. Let the parent chromosomes are A and B. Then the Child-1 is obtained by having A as Parent-1 and B as Parent-2 and Child-2 is obtained by having B as Parent-1 and A as Parent-2.

Bad Population repository:
In order to determine weak chromosomes and eliminate them a repository is maintained. The repository is a size of $\lfloor \sqrt{N} \rfloor$, where $\lfloor x \rfloor$ defines maximum integer less or equal to x.

1. In every iteration any 2 chromosomes are selected randomly and gone through crossover and mutation and the offspring competes with the chromosomes of main population. If these

offspring are fitter than the worst chromosomes of the main population then the offspring substitutes the worst chromosomes of main population.
2. In every iteration one chromosome is selected randomly and crossed over with any randomly selected chromosome from main population. This offspring also competes with the chromosomes of main population for its existence.

No updating of chromosome happens here.

Mutation:
Mutation is applied to avoid local optimum.
Here we have taken mutation probability 0.8.

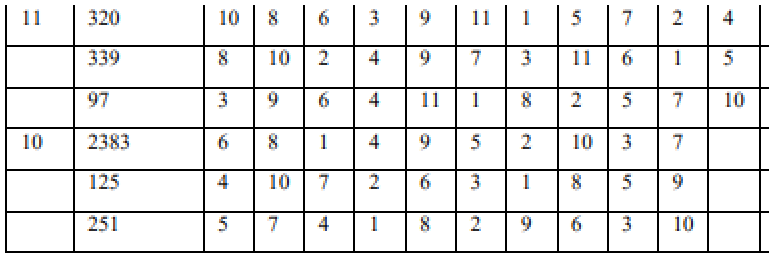
Before mutation: 2 6 8 3 4 1 7 5
After mutation: 2 6 7 3 4 1 8 5

In a single mutation function we are randomly select two genes and swap them. If this process is carried out twice it is called Double mutation. Probability of double mutation happening is 0.4

2. Results observed by the authors.

The proposed Genetic Algorithm provides 3 solutions for all values of N.
It takes fairly less amount of time for 2 solutions then those approaches proposed before.

Results:
N Iteration Solution



11	320	10	8	6	3	9	11	1	5	7	2	4
	339	8	10	2	4	9	7	3	11	6	1	5
	97	3	9	6	4	11	1	8	2	5	7	10
10	2383	6	8	1	4	9	5	2	10	3	7	
	125	4	10	7	2	6	3	1	8	5	9	
	251	5	7	4	1	8	2	9	6	3	10	

3. Observations and Conclusion:

This paper reveals that the N-Queens problem can be solved within a reasonable time by using metaheuristics like Genetic Algorithm and its modified forms. N-Queens Problems have a modicum of practical usage yet they represent a major class of NP problems that cannot be solved in a reasonable amount of time using deterministic methods- the reason of their importance.
In the previous literatures, attempts were made to solve N-Queens Problems with Genetic Algorithms. In this paper with modifications in the fitness functions, hereby giving rise to a novel modified GA, it is seen that this approach yields promising and satisfactory results in less time compared to that obtained from the previous approaches.