Report Question 2

N-Queen's Problem using Genetic Algorithm

Question 2

a) Solve the 8-queens problem using genetic learning algorithm

b) Extend the above to n-queens problem

Problem Statement

8- Queen's problem is a classical problem in which we are required to place 8 queens on an 8*8 chessboard in such a way that no two queens are able to attack each other. A legal move by a queen include any number of steps row wise or column wise or diagonal wise.

An N-queen problem is generalization of 8-queen problem such that N non-attacking queens can be placed on N*N chessboard.

Chess composer Max Bezzel published the eight queens puzzle in 1848. Franz Nauck published the first solutions in 1850. Nauck also extended the puzzle to the N-queens problem, with N-queens on a chessboard of N*N squares. (source:Wikipedia)

Approach-Algorithm

Here, the idea of Genetic Algorithmic approach is used to solve this problem based on following algorithm.

```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
    new_population ← empty set
    for i = 1 to SIZE(population) do
      x ← RANDOM-SELECTION(population, FITNESS-FN)
      y ← RANDOM-SELECTION(population, FITNESS-FN)
      child ← REPRODUCE(x, y)
      if (small random probability) then child ← MUTATE(child)
      add child to new_population
    population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN
```

Assumptions and Methodology

Chromosome: [a1……an] for n = Number of queen

For example : [5,3,6,0,7,1,4,2] is a chromosome for 8 queen problem where n=8 and values in the chromosomes are in range of 0 to 7.

In this representation, index of list are rows and value at index is the column where the queen is placed and can be viewed as:

0	1	2	3	4	5	6	7
					Q		
			Q				
						Q	
Q							
							Q
	Q						
				Q			
		Q					

Population: List of all chromosomes.

Fitness Function: Number of pairs of queens in attacking position for a state.

Parent Selection: Top K parents are seleted.

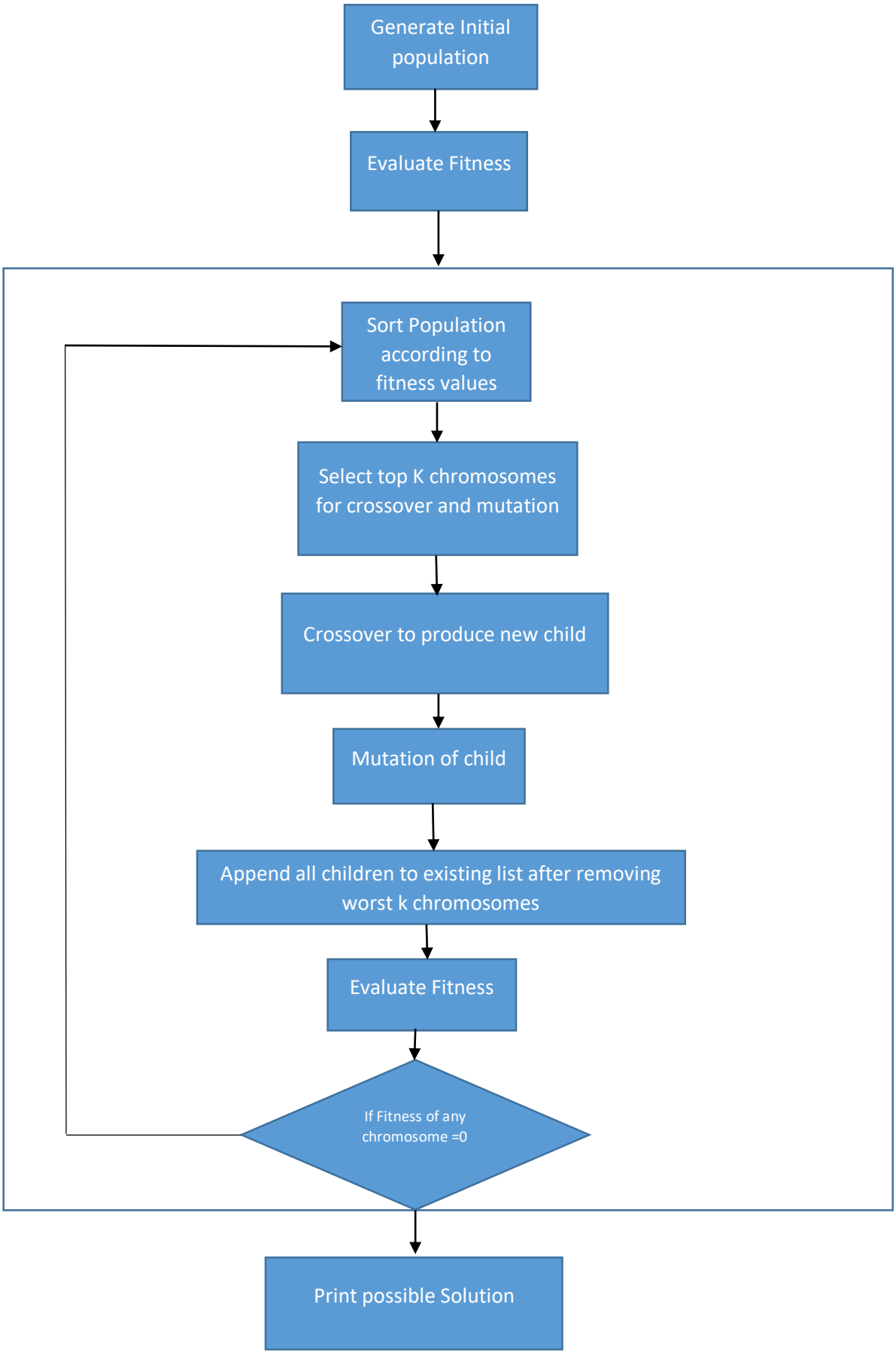
Crossover/Reproduction: Taking alternate values from the state of the board.

Mutation: Random change in one position on queen in the crossover state.

Break condition for loop : Whenever a state is found with fitness value =0 i.e. no attacking pair of queen's in the state.

Output: Solution to N-queen's problem and number of iteration or new generations used to have a solution.

Code Flow


Generate Initial population
Evaluate Fitness
Sort Population according to fitness values
Select top K chromosomes for crossover and mutation
Crossover to produce new child
Mutation of child
Append all children to existing list after removing worst k chromosomes
Evaluate Fitness
If Fitness of any chromosome =0
Print possible Solution

Inferences and Conclusion

Example Run 1 :

N=8 #Number of queens

p=20 #population size

k=10 #Selecting k best states

output:

```
Found solution
[5, 3, 6, 0, 7, 1, 4, 2]
Total iterations =
150
```

Example Run 2:

N=8 #Number of queens

p=20 #population size

k=6 #Selecting k best states

output:

```
Found solution
[0, 5, 7, 2, 6, 3, 1, 4]
Total iterations =
86
```

Example Run 3:

N=14 #Number of queens

p=20 #population size

k=10 #Selecting k best states

output:

```
Found solution
[0, 6, 4, 11, 8, 12, 5, 3, 1, 10, 13, 2, 9, 7]
Total iterations =
6095
```

Example Run 4:

N=20 #Number of queens

p=100 #population size

k=10 #Selecting k best states

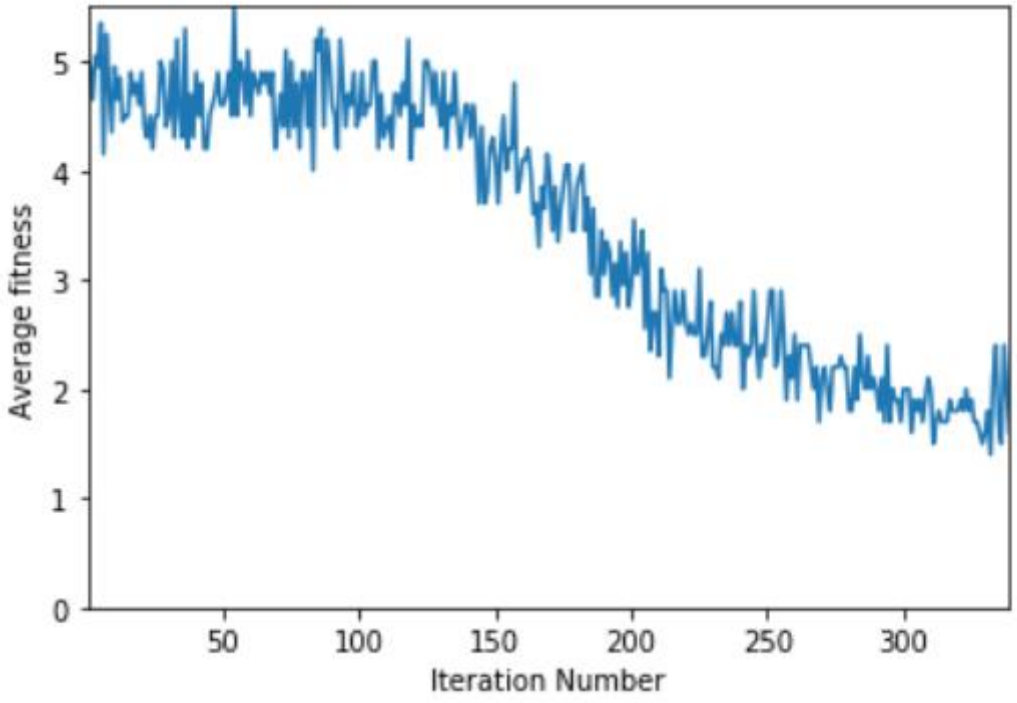
```
Found solution
[1, 12, 16, 5, 0, 17, 11, 14, 7, 3, 19, 9, 2, 13, 18, 4, 8, 10, 15, 6]
Total iterations =
11839
```

Therefore,

1. Number of queens, population size and selecting k states for crossover plays and important role in finding a solution.
2. Exponential rise in the number of iteration when N is increased.

Average Fitness Value per iteration:

Low average fitness value is good in this case. We can see that average fitness value keeps on fluctuating up and down due to the randomness in our algorithm but with time it decreases till the algorithm converges.



Note: One code is submitted which has a global variable N for number of queens and can be changed according to the question since extending a solution was asked in part b.

References:

Wikipedia

Book-AI by Russell and Norvig, 3rd Edition

Python tutorials