Department of Computer Science and Engineering

Course Code:CSE422

Course Name: Artificial Intelligence **Prerequisite:** CSE111, CSE221

Lab 03
Genetic Algorithm

I. **Lab Overview:**

The students will solve N-Queen problem using python programming and visualizing the evolution performance.

II. **Learning Objective:**

a. Introducing the 4-Queen problem
b. Solution of 4-Queen problem in Backtracking approach
c. Demerits of Backtracking approach
d. Introducing 8-Queen problem
e. Discussion on Genetic Algorithm
f. Solution of 8-Queen problem using GA

III. **Lesson Fit:**

There is pre-requisite to this lab: CSE111, CSE221. You should have intensive Programming Knowledge and capability to understand algorithms.

IV. **Acceptance and Evaluation**

Students will show the output using different datasets and python code. They will be marked according to their lab performance. The main evaluation criteria will be based on project report and demonstration.

V. **Learning Outcome:**

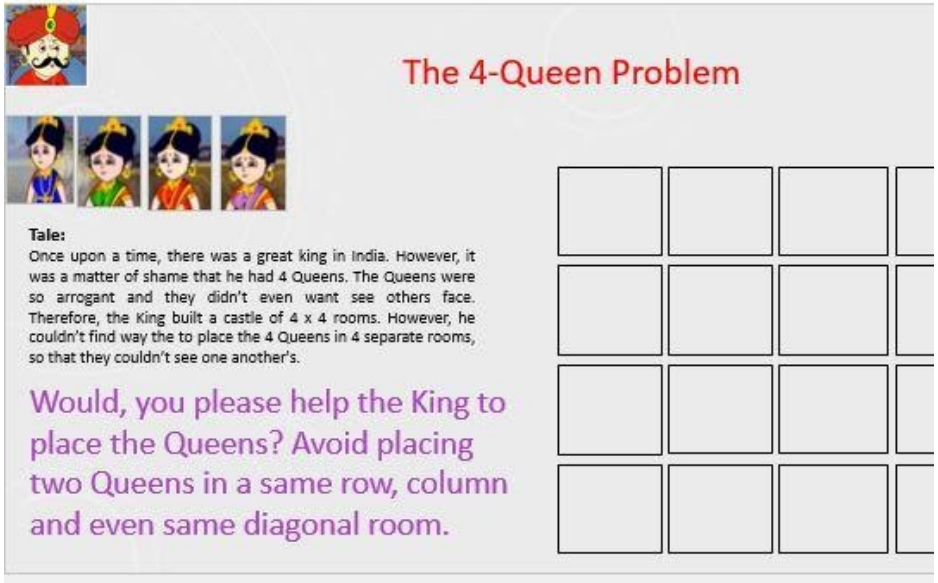
After this lab, the students will be able to:

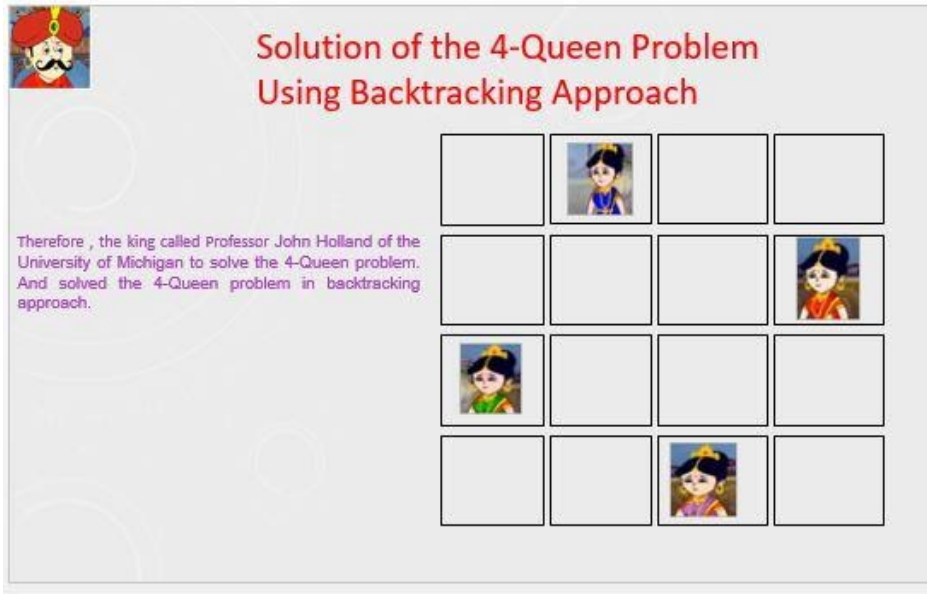
g. Demerits to solve N-Queen problem using Backtracking approach.

h. Solve the N-Queen problem using Genetic Algorithm

VI. Activity Detail

- **Hour: 1.0 - 2.0**





The 5-Queen Problem

One month later, Professor received a call from the great King to solve his 5-Queen problem. Professor, solved the 5-Queen problem in backtracking approach.

Solution of the 5-Queen Problem Using Backtracking Approach

6-Queen Problem

John Holland introduced Genetic Algorithm (GA)

Darwin's theory of evolution

Fortunately, one month later, the King requested the professor to solve 6-Queen problem. The professor thought that the King may request him to solve 16-Queen problem within next 10 months.

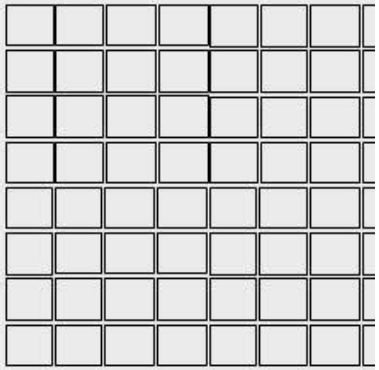
Backtracking approach will not be efficient to solve the 8 or 16-Queen problems.

Therefore, professor invented Genetic Algorithm to solve the n-Queen problem.

8-Queen Problem

John Holland introduced Genetic Algorithm (GA)

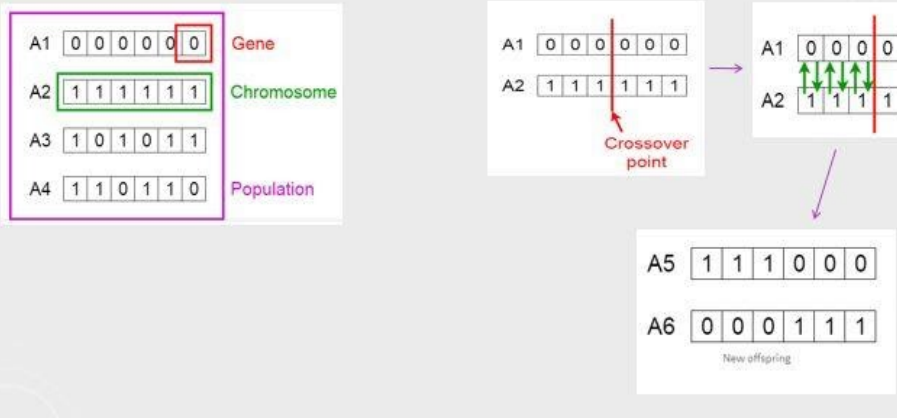
Darwin's theory of evolution



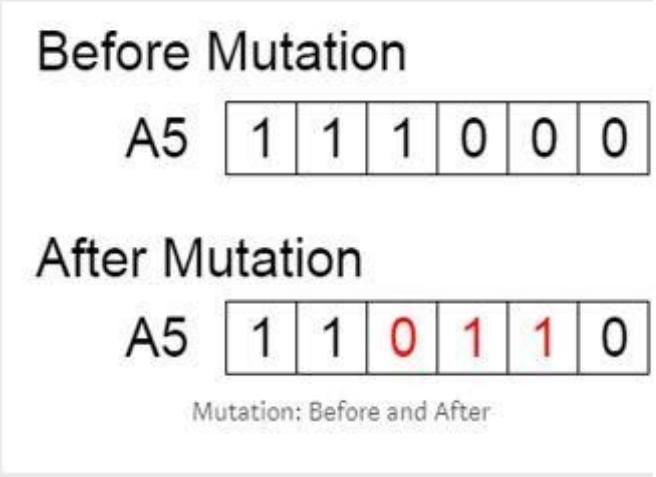
Introduced in the 1970s by John Holland at University Michigan

- begin with *k* randomly generated states (populatic
- each state (individual) is a string over some alpha (chromosome)
- fitness function (bigger number is better)
- crossover
- mutate (evolve?)

Crossover:



Mutation:

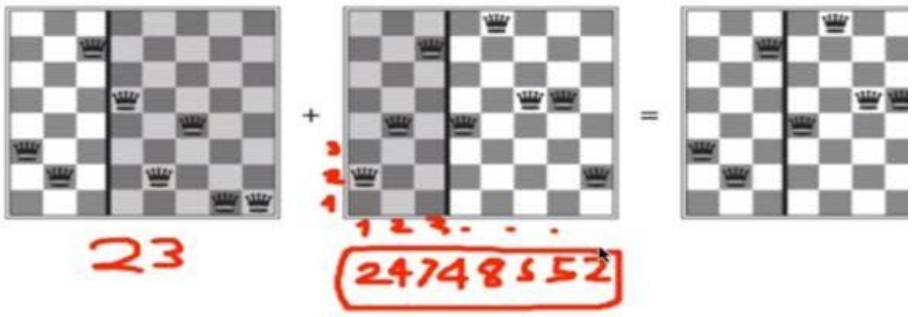


Pseudo-code of GA:

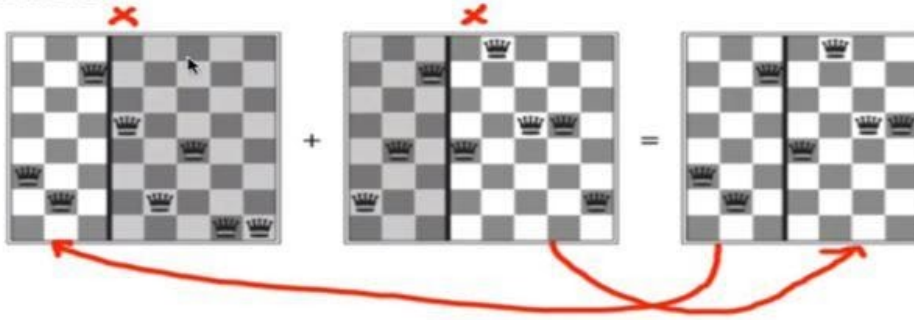
START
    Generate the initial population
    Compute fitness
    REPEAT
        Selection
        Crossover
        Mutation
        Compute fitness
    UNTIL population has converged
STOP

Fitness Function: Pairs of nonattacking queens

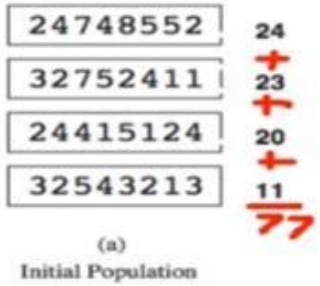
That way, higher scores are better.

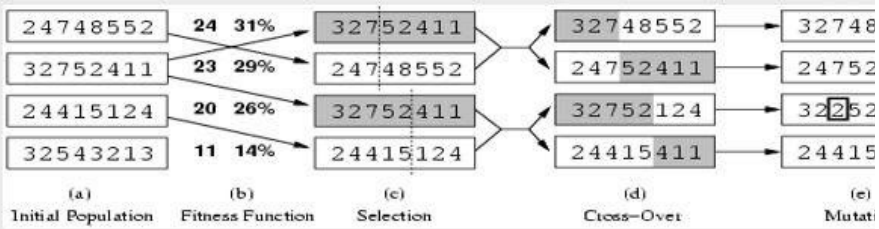


The good genes (features) of the parents are passed ont
children

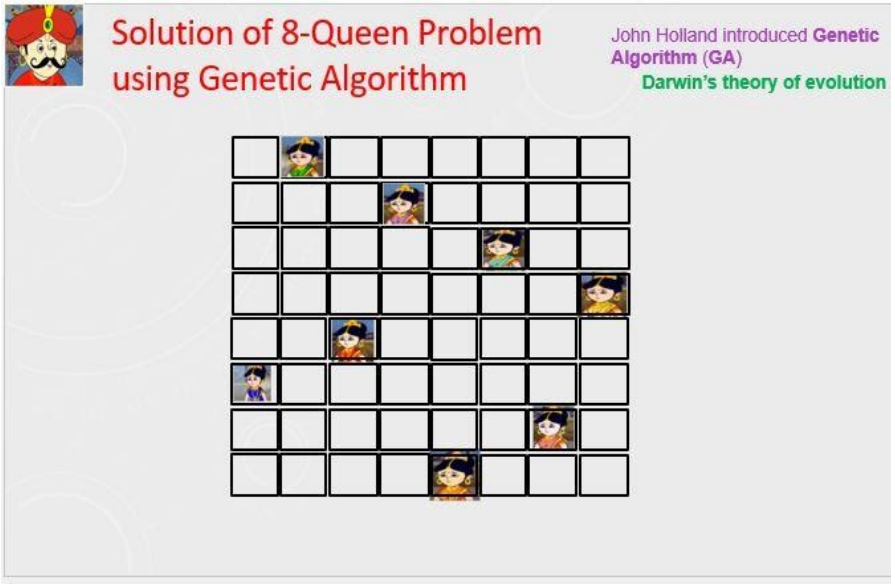


Represent states and compute fitness functio





- Fitness function: number of non-attacking pairs of queens (min = 0, max = 8 × 7/2 = 28)
- 24/(24+23+20+11) = 31%
- 23/(24+23+20+11) = 29% etc



Application areas of Genetic Algorithm:

- Game programming
- Cloud resource allocation
- Job scheduling of operating systems
- Channel assignment in communication sys
- Combinatorial optimization
- Integer programming
- operational research

- **Hour: 2.0-3.0**

(It is Not a Group Task, Try Individually)

Marks: 10 **Time: 50 minutes**

Task 1: Implement N-Queen problem using Genetic Algorithm in python programming.

Task 2: Visualize the evolution through plotting the changes of fitness values, and the variances of fitness values for convergence.

Hints: Take help from Prateek Joshi's Book chapter 8, you can follow Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

Evaluation Process (VIVA and Written answers): You have to explain your program and show your work to the Lab Instructor. Instructor may ask you some questions to evaluate your knowledge and expertise level.

---End---