Presented by: krapa

**KRAPALI RAI**

**SIDDHANT SINGHAL**

Classroom Schedule Generator

Using Genetic Algorithms

PrOJECT Report

April 15, 2017

# **Classroom Schedule Generator**

## **PROBLEM Statement:**

In real world, a potential scenario where Genetic Algorithm can be implemented is generation of efficient time schedules. Efficient schedule generator finds its application in various problem for example, Classroom scheduling, Flight landing schedule problem, Job scheduling and many more.

We have considered a problem where we need to come up with a semester timetable for classes in an University for a particular department. Our task is to generate an optimum schedule where there is least possible or no conflicts between classes. We have defined conflicts as follows:

CONFLICTS:

>A professor is teaching more than one class at the same time.

>A room is hosting more than one class at the same time.

>A room conducting a class which has more number of student than the capacity of the class.

### **IMPlementation Design:**

The problem is to find an optimum set of schedules for N number of lectures that are need to be conducted in given K number of classrooms. The problem arises when we have lesser number of rooms and more number of sessions(lectures) to be conducted in a given constraint of time. Following are the GA implementation approach:

*Organism:*

The potential solution(lecture) that consists of set of genotype (professor, meeting time, classroom) and phenotype environmental characteristics.

*Genetic code:*

Genes are the heritable genetic identity. In our problem we consider the genotypes (set of genes) as Classroom, Meeting time, Professor conducting a lecture in the given classroom at given time. Our phenotype is the lecture is generated by exploiting these genes. Implementing crossover, mutation and elitism in our genetic code provide us have a fair chance to find an optimal schedule set while considering all possible gene permutation.

*Gene expression:*

In order to observe the effect of the gene variation on an organism (individual lecture) we are considering an set of lectures that can be conducted in the classrooms with or without conflicts. The expression is generation a set of schedules which have been implemented using Arraylist of lectures and sorted by their level of fitness.

*Fitness function:*

Schedules are sorted by fitness function. Fitness level in our problem is based on the minimum possible number of conflicts. Our fitness function is inversely proportional to the number of conflicts encountered.

*Mutation:*

To inject variations among the population mutation has been used. The value of mutation has been kept low in order to perform better selection by Elitism.

*Crossing Over:*

Considering a fixed number of Elite population that will reappear in next generation and performing crossover for best population to generate offspring in order to obtain better results for next generation.

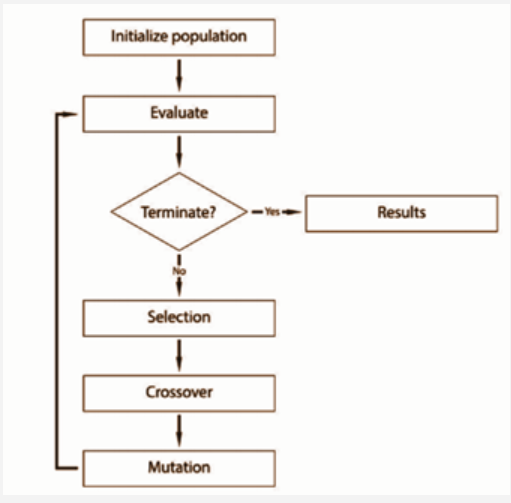
*Evolution:*

By performing Crossover Culling and Mutation, we are performing Evolution from one generation to other generation. After crossover the population of new generation will be sorted by a fitness function. Hence providing fair chance for getting new candidates(schedules) in the elite population in order to achieve a better solution.

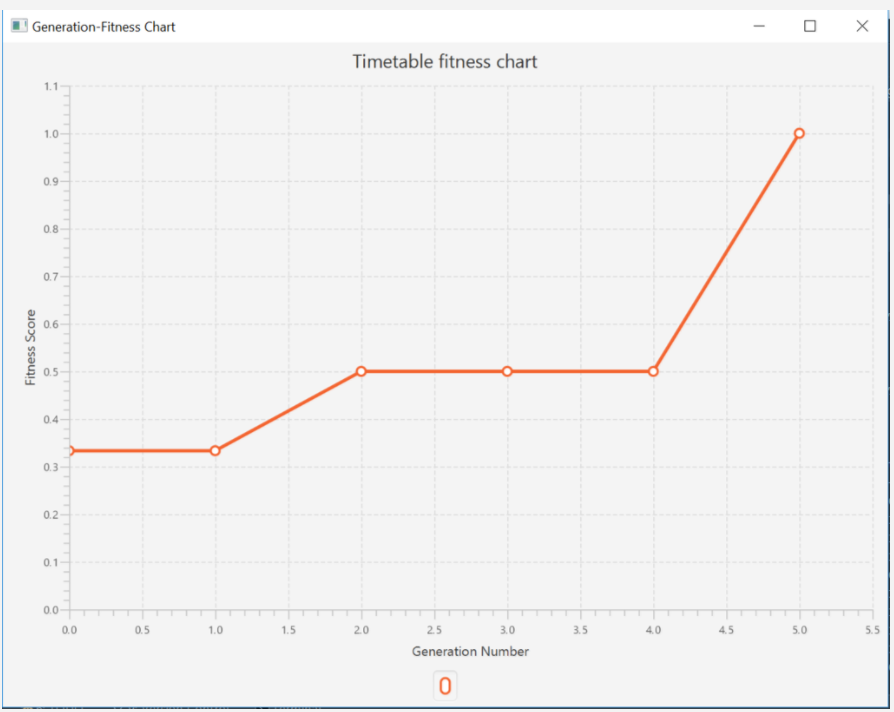
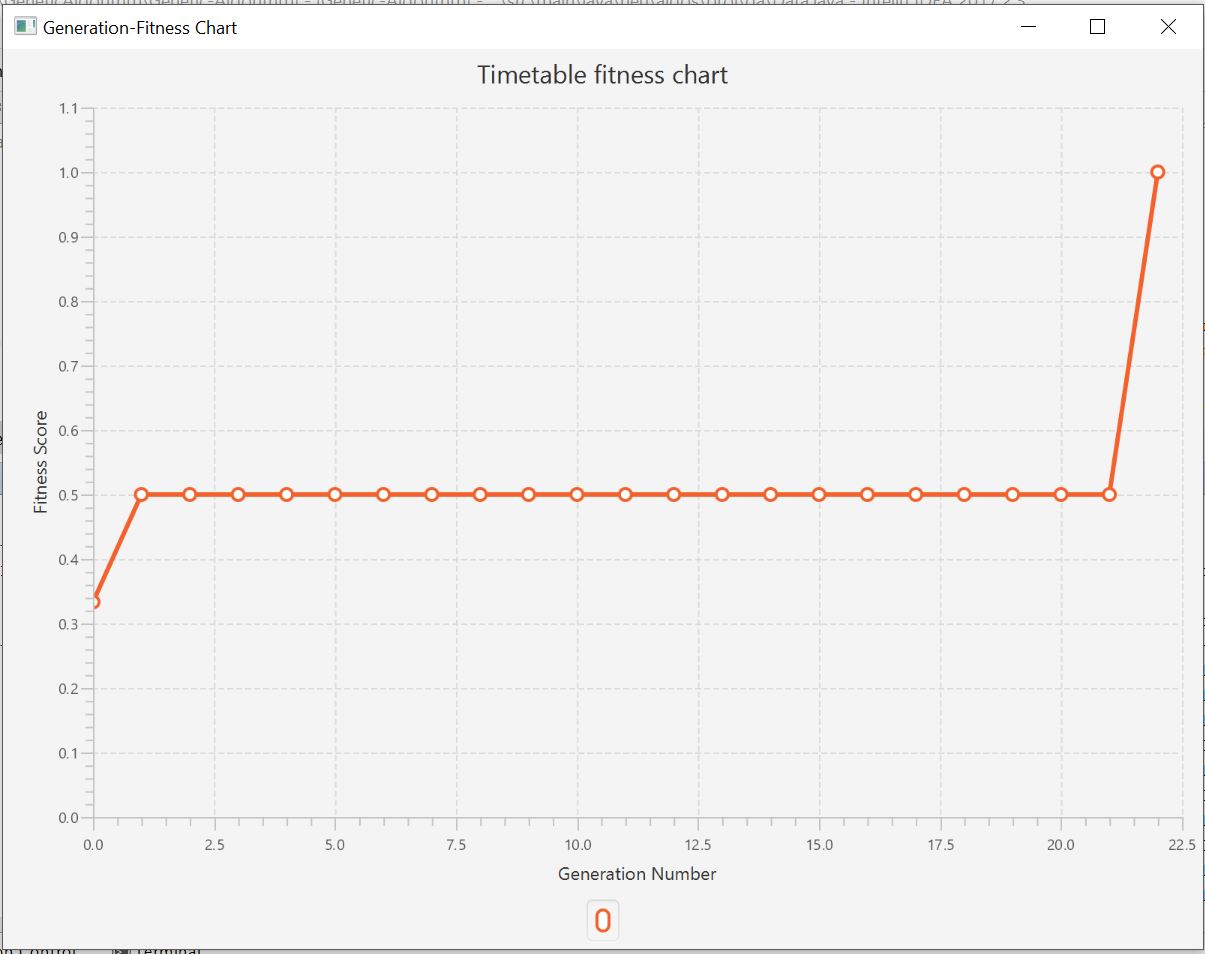
*Elitism:*

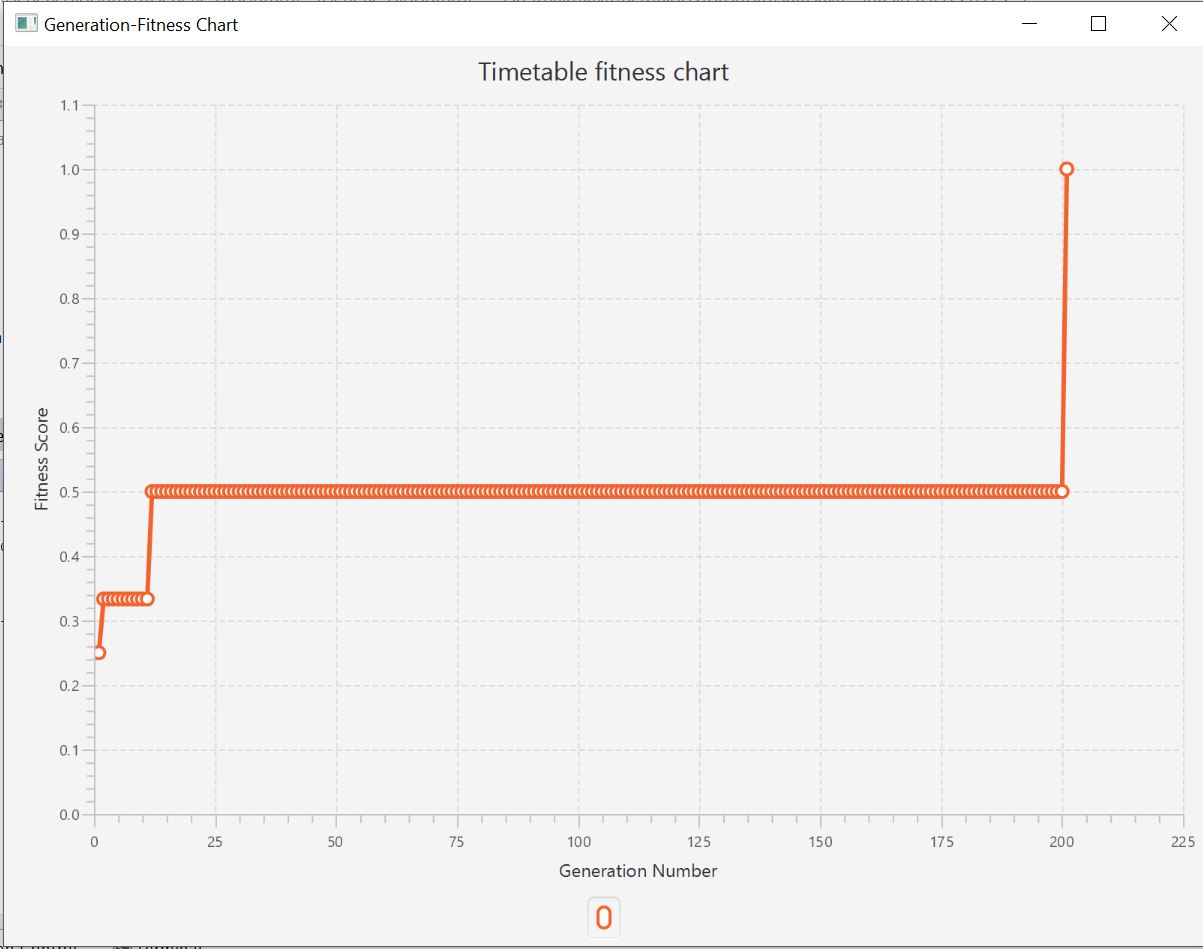
We are selecting a list of fit population in a particular generation that is going to be a part of crossover in order to generate better schedule set.

**Flow of Evolution:**

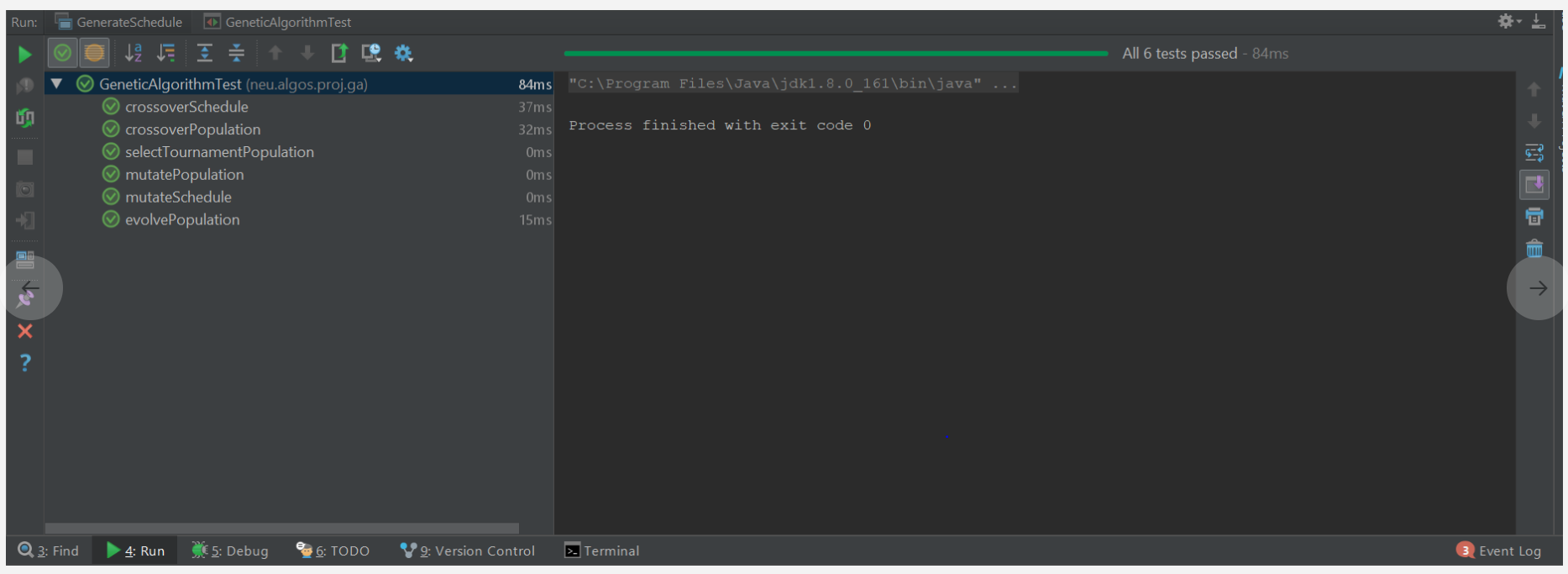


**RESULTS:**





**TESTCASES:**



Output:

