**CHAPTER 3: PROPOSED SYSTEM**

**3.1 Architecture Framework of Proposed System:**

Timetable is a very common problem when it comes to Scheduling and that is time consuming, energy sapping and leads to waste of resources. Almost all Universities have issues when it comes to lecture scheduling due to all the constraints such as lecturer’s availability, classroom and lab availability, etc.

An efficient way for scheduling this is using Genetic Algorithm. Genetic Algorithms are adaptive heuristic search/optimization algorithms that belong to the family of evolutionary algorithms. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection. It is based on the concept of survival of the fittest.

The algorithm can be summarised as:

* + 1. A population p is randomly initialised
    2. The fitness of the population is determined
    3. Until the stopping criterion is met, do:
       1. **Select** parents from population
       2. **Crossover** and generate new population
       3. Perform **mutation** on new population
       4. Calculate fitness of new population

The main operators of GA are:

1. **Selection**: It is the process of selecting a part of the population to breed a new generation.

Figure 3.1 shows Roulette Wheel Selection which is a selection procedure in which the possibility of selection of a chromosome is directly proportional to its fitness.

Figure 3.1 Roulette Wheel Selection

1. **Crossover**: Also known as recombination, it makes use of two parents’ properties to generate a child.

Figure 3.2 shows Two Point Crossover



Figure 3.2 Two Point Crossover

1. **Mutation**: The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence

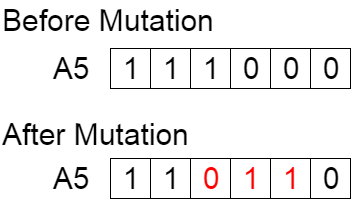


Figure 3.3 Mutation

Now that we have discussed the main steps in Genetic Algorithm, let us outline the framework of the proposed system.

We will make use of a 2D matrix called a ‘Target Matrix’. Along with this matrix, six sets with are pertinent to timetable scheduling will be defined. These six include:

1. Course Code set **M** = {M1, M2, M3, …}
2. Lecturer Code set **L** = {L1, L2, L3, …}
3. Class Code set **C** = {C1, C2, C3, …}
4. Room Code set **R** = {R1, R2, R3, …}
5. Day set **D** = {D1, D2, D3, …}
6. Hour set **H** = {H1, H2, H3, …}

Each column in the matrix will represent a set of <**M, L, C**> which represents Lecturer ‘**L**’ teaching Course ‘**M**’ for Class ‘**C**’.

Each row in the matrix will represent a set of <**R, D, H**> which represents Room ‘**R**’ on Day ‘**D**’ at Hour ‘**H**’.

For set ‘X’, N(X) is defined as number of elements of X.

Number of columns however would be the number of valid combinations of <M, L, C>.

Target Matrix:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | m1, l1, c1 | m1, l2, c1 | m2, l1, c2 | … |
| r1, d1, h1 | 1 | -1 | -1 | … |
| r1, d1, h2 | 1 | -1 | -1 | … |
| r1, d2, h1 | -1 | 1 | -1 | … |
| r1, d2, h2 | -1 | 1 | -1 | … |
| r2, d1, h1 | 1 | -1 | -1 | … |
| r2, d1, h2 | 1 | -1 | -1 | … |
| r2, d2, h1 | 0 | 0 | 0 | … |
| r2, d2, h2 | -1 | -1 | 1 | … |
| … | … | … | … | … |
| No. of units Scheduled (th) | 4 | 2 | 1 |  |

Table 3.1 Target Matrix

From the target matrix for an element Mij, if it is set to 1, it indicates that Lecturer ‘L’ teaching Course ‘M’ for Class ‘C’ is scheduled to Room ‘R’ on Day ‘D’ at Hour ‘H’.

Each Cell value can have the following values

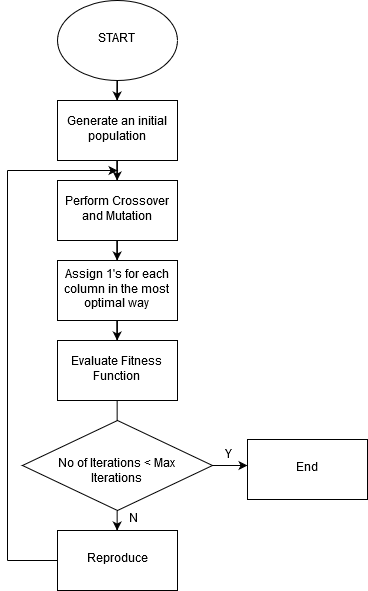
1. Mij = 0, indicates that the slot is available to be scheduled
2. Mij = 1, indicates that the slot has been scheduled
3. Mij = -1, indicates that the slot cannot b scheduled

Also, for each element Mij a set of 6 functions are defined namely fm, fl, fc, fr, fd and fh which return information of attributes course, lecturer, class, room, day and hour respectively.

Apart from the above, certain rules must be followed:

1. , i.e. sum of all non-negative numbers in a row should be less than or equal to one.
2. , i.e. the number of hours that a combination of <M, L, C> is scheduled should be the number of hours required for it.
3. The combination <M, L, C> should only be scheduled in combination <R, D, H> if it can be accommodated in R.

The Flowchart 3.1 indicates the proposed Algorithm



Flowchart 3.1 Proposed Algorithm

Genetic Algorithm:

1. Gene Representation: The chromosome will be represented as a string of numbers made up of the column ids. For e.g. an 8 Gene representation can be {4, 5, 2, 1, 3, 8, 6, 7}. In this case we shall arrange the 4th column first then the 5th and so on.
2. Initial Population: An Initial Population will be randomly generated. In the above example we looked at one chromosome, a population would be a collection of chromosomes that are mostly unique. A population example is C1 = {4, 5, 2, 1, 3, 8, 6, 7}, C2 = {4, 7, 8, 2, 3, 1, 6, 5}, …, Cn.
3. Crossover: A two-point Crossover will be used wherein parents are divided into three portions cut by two randomly generated points. Matching sections of two individuals are swapped to create offspring. Example

1st Parent: {4, 5, 2, 1, 3, 8, 6, 7}

1st offspring: {4, 5, 8, 2, 3, 1, 6, 7}

2nd offspring: {6, 7, 2, 1, 3, 8, 4, 5}

2nd Parent: {6, 7, 8, 2, 3, 1, 4, 5}

1. Mutation: Two random Genes are swapped while inversion.
2. Fitness Evaluation: Let be number of units scheduled in kth column and p is the maximum column in target matrix, means unit number of jth course to be scheduled for each course and t is total classes. Fitness is given by:

**3.2 Details of Hardware and Software:**

Technologies that will be used for the implementation:

* + Backend Development
    - PHP
    - Java 8
  + Database
    - MySQL
  + Frontend Development
    - React JS
  + Logical Implementation
    - J2EE Technologies/Spring Framework

**3.3 Design Details:**

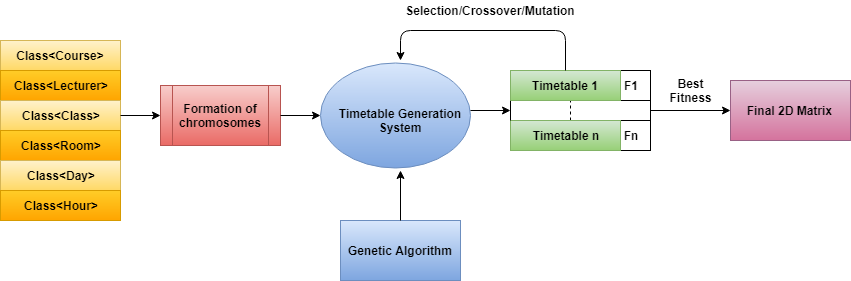


Figure 3.4 Proposed Architecture

As Discussed in the Framework, the Target Matrix makes use of six sets. Each Element in a set will be an Object which will be identified uniquely by an ID. The following are the description of each Object:

1. The **Class** has an ID, name of student group, number of subjects, array of subject names and hours of study required for each subject per week. It also contains the id of teachers who will teach those subjects.
2. The **Lecturer** is a class to hold the faculty information. It an id name faculty, subject that he/ teaches and an in assigned which represents the no. of batches assigned to the teacher.
3. The **Course** is a class that hold the information on the Course, the Faculty that teach it, whether it is a lab or regular lecture and is identified by Course ID.
4. The **Room** is class which stores the details of a classroom/lab, its occupancy, its speciality, corresponding courses to be held, venue details and is identified by Room ID
5. The **Hour** class represents the timeslots available for Scheduling.
6. The **Day** class represents the Days that the Lectures are scheduled on.

**3.4 Methodology:**

The given problem is an NP hard problem with no specific approach to solve it. We will be using genetic algorithm to solve the problem based on our previous study which highlighted the superiority of genetic algorithm over other algorithms.

The given application has four main steps:

1.Taking input from the user

* This indicated various constraints that would have to be considered for generating the timetable.
* The information would contain details about the students, teachers, classrooms, labs, days and lecture hours.
* The above information would be provided to the application through a user interface designed to do so.

2.Mapping the input to genetic algorithm.

* The input received from the user end will be stored in the form of objects.
* However, genetic algorithm cannot work on objects it needs the input to be in the form of chromosomes.
* This step will consist of converting all the inputs to chromosomes.

3.Generating the timetable

* The final deliverable of the project is the timetable with the best fitness.
* Genetic algorithm will perform the operations like selection, crossover and mutations iteratively on the chromosomes to generate the timetable.
* At each step the fitness of each timetable generated will be calculated.
* The application will return the timetable which has the highest fitness score.

4.Final timetable

* The final timetable is a 2D matrix which contains the schedule for the given institution which has the maximum fitness.
* The timetable must be sent to the given organization in the format that they would approve.
* E.g. XLS, text, excel, etc.

So, the final step is returning the delivering the final timetable in any suitable format as requested by the given institution.