A Project Report
on

Institute Timetable Scheduler

by

1. Giriraj Belanekar
2. Bhaven Gore
3. Disha Shirdhankar

under the guidance of

Prof. Sunil.P.Khachane

Department of Computer Engineering

University of Mumbai

April - 2020

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53.

Certificate

Department of Computer Engineering

This is to certify that

1. Giriraj Belanekar
2. Bhaven Gore
3. Disha Shirdhankar

Have satisfactory completed this synopsis entitled

Institute Timetable Scheduler

Towards the partial fulfillment of the
BACHELOR OF ENGINEERING
IN
(COMPUTER ENGINEERING)
as laid by University of Mumbai.

Guide	**Head of Department**
Prof. Sunil.P.Khachane	**Dr.Satish Y. Ket**

Principal
Dr.Sanjay Bokade

Internal Examiner	**External Examiner**

Project Report Approval for B. E.

This project report entitled **Institute Timetable Scheduler** by **Giriraj Belanekar, Bhaven Gore, Disha Shirdhankar** is approved for the degree of Computer Engineering.

Examiners

1. ———————————

2. ———————————

Date:

Place:

Declaration

We wish to state that the work embodied in this synopsis titled "Institute Timetable Scheduler" forms our own contribution to the work carried out under the guidance of "Prof. Sunil.P.Khachane" at the Rajiv Gandhi Institute of Technology.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. we understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Giriraj Belanekar (712)

Bhaven Gore (730)

Disha Shirdhankar (731)

Abstract

The manual system of preparing time table in colleges with large number of students is very time consuming and usually ends up with various classes clashing either at same room or with same teacher having more than one class at a time. To overcome these problems we propose to make an automated scheduler. The system will take various inputs like details of students, subjects and classrooms, teachers available, depending upon these inputs it will generate a possible time table. Making optimal utilization of all resources in a way that will best suit any of constraints or college rules. For a solution to be feasible, a number of hard constraints must be satisfied. The results obtained confirms that the approach is able to produce solutions to the Institute timetabling problem which exhibit some of the lowest penalty values in the literature on these benchmark problems. This problem Statement can be achieved using Genetic Algorithm.

Contents

List of Figures

List of Algorithms

List of Acronyms

- **Chromosome:** In an obvious nod to biology, a chromosome is a single hypothesis of which many make up a population.
- **Population:** In a GA, each iteration, or generation, results in a series of possible hypotheses for best approximating a function, and the population refers to the complete set or pool of these generated hypotheses after a given iteration.
- **Generation:** A generation, then, is simply the full set of the results of a GA iteration.
- **Fitness:** We need some metric to measure the best fit of a hypothesis.
- **Crossover:** Just like in biological terms, mutation is used in GAs in order to push hypotheses toward optimal. Generally used sparingly, mutation would simply flip the bit of a random gene and push the entire chromosome forward to the subsequent generation, a strategy for escaping potential local minima.

Chapter 1

Introduction

1.1 Introduction Description

The manual lecture-timetable scheduling demands considerable time and efforts. Even though most college administrative work has been computerized, the lecture timetable scheduling is still mostly done manually due to its inherent diculties.

The lecture-timetable scheduling is a constraint satisfaction problem in which we nd a solution that satises the given set of constraints. Since the problem is a combinatorial optimization problem belonging to NP-hard class , the computation time for timetabling tends to grow exponentially as the number of variables increase. In our project, timetable problem is formulated as a constraint fulfillment problem and we proposed a realistic timetable algorithm which is capable of taking care of both hard and soft constraints.

1.2 Organization of Report

Describe every chapter (what every chapter contains)

- **Ch.1 Introduction:** An approach to generate an Automatic timetable,which will reduce manual efforts and save time; making it more effective and efficient.
- **Ch.2 Review of Literature:**Difficult to write a universal agenda, fitting for all possible timetable problems. Even though manual creation of timetable is sustained, it is still universal, because of the lack of suitable computer programs.
- **Ch.3 Proposed System:**Earlier systems used Dynamic problem making it a NP Hard problem. This system aims to generate the time table in polynomial time.
- **Ch.4 Implementation Plan:**Collecting the data of all the teachers, subjects, classrooms, labs etc and generating a time table by avoiding clashes.
- **Ch.5 Conclusion:**It is complicated task that to handle many Faculty's and allocating subjects for them at a time physically. So our proposed system will help to overcome this disadvantage.

Chapter 2

Literature Review

There exist various problems such as Sports Timetabling,Examination Timetabling, Employee Timetabling and university timetabling. Carter and Laporte (1998) considered different categories to solve the timetabling problem. They are – Cluster method, Sequential method, Meta-Heuristics and Constraint Based method. Meta Heuristics is a higher level procedure which is used to provide good enough solutions for optimization problems. On some class of problems, they do not guarantee a globally optimum solution. This method is used when the classical methods are too slow or fail to give a solution. This is achieved at the cost of optimality and precision for speed.

2.1 Survey Existing system

The first paper proposed that finding a feasible lecture/tutorial timetable in a large university department is a challenging problem faced continually in educational establishments. This paper presents an evolutionary algorithm (EA) based approach to solve a heavily constrained university timetabling problem. The approach uses a problem-specific chromosome representation. Heuristics and context-based reasoning have been used for obtaining feasible timetables in a reasonable computing time. An intelligent adaptive mutation scheme has been employed for speeding up the convergence. But this system is difficult to implement since it considers entire university problem and evolutionary algorithm. [1].

Second paper proposed that The university course timetabling problem (UCTP) is a combinatorial optimization problem, in which a set of events has to be scheduled into time slots and located into suitable rooms. The design of course timetables for academic institutions is a very difficult task because it is an NP-hard problem. This paper investigates genetic algorithms (GAs) with a guided search strategy and local search (LS) techniques for the UCTP. The guided search strategy which is used here is to create offspring into the population based on a data structure that stores information extracted from good individuals of previous generations. The LS techniques use their exploitive search ability to improve the search efficiency of the proposed GAs and the quality of individuals. The proposed GAs is tested on two sets of benchmark problems in comparison with a set of state-of the- art methods from the literature. The experimental results show that the proposed GAs is able to produce promising results for the UCTP. [2].

It has developed in three modules 1) Insertion Module In this module we provide various user inputs to our system which acts raw data for creating the final time table. 2) Allocation Module. In this module, user can choose any semester randomly to start the process. He

starts filling the slots from the Monday by selecting the particular subjects, faculty that is mapped with that subject gets allotted to that slot of the day. The various soft and hard constraints are checked every time the slot is filled. It will not be blocked if any of the constraints is not satisfying.We are checking all these constraints by writing query in stored procedure and using the database data 3) Display Module In this module we can view how the time table is generated of each class. We have also provided the feature to view the class time table and faculty wise time table [3]

2.2 Limitation Existing system or research gap

- All the Existing methods used where either manual, local search or using algorithms whose time complexity reaches polynomial time.
- As mentioned, when Timetable generation is done, it should consider the maximum and minimum workload that is in a college.In those cases, timetable generation will become more complex. Also, it is a time consuming process.
- As we know all Institutions or organizations have their own timetable; managing and maintaining these will not be difficult. Considering workload with this scheduling will make it more complex.

2.3 Problem Statement and objectives

Our project entitle "Institute Timetable scheduler" is meant to generating timetable scheduling process in colleges or in any other institutions which could minimize the human effort and maximise the efficiency and the timetable is stored in a excel sheets.

2.3.1 Objectives

List objectives using bullet points

- The final system should be able to generate time tables in completely automated way which will save a lot of time and effort of institute administration.
- To make the timetable system generic so that it can work equally well for different schools, colleges and universities.
- User defined constraints handling.
- Ease of use for user of system so that he/she can make automatic timetable..
- Focus on optimization of resources i.e teachers, labs and rooms etc
- Provide a facility for everyone to view the timetable.
- Generate multiple useful views from timetable

2.4 Scope

Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

Chapter 3

Proposed System

The problem involves assigning lecture events to timeslots and rooms subject to a variety of hard and soft constraints. Hard constraints represent an absolute requirement. A timetable which satisfies the hard constraints is known as a feasible solution. We will test our approach on the problem instances for following constraints:

- No student can be assigned to more than one course at the same time.
- The room should satisfy the features required by the course.
- The number of students attending the course should be less than or equal to the capacity of the room.
- No more than one course is allowed at a time slot in each room.
- A student has more than 2 consecutive courses.
- A professor has a single course on a day

3.1 Analysis/Framework/Algorithm

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

3.2 Details of Hardware and Software

3.2.1 Software Requirements

- OS: MS windows XP or above / Ubuntu
- Framework: Python tkinter
- Database: sqlite3 , mysql

3.2.2 Hardware Requirements

- Processor: Pentium Dual core or higher
- RAM: 512 MB or higher
- Hard disk: 2 GB or higher

3.3 Design Details

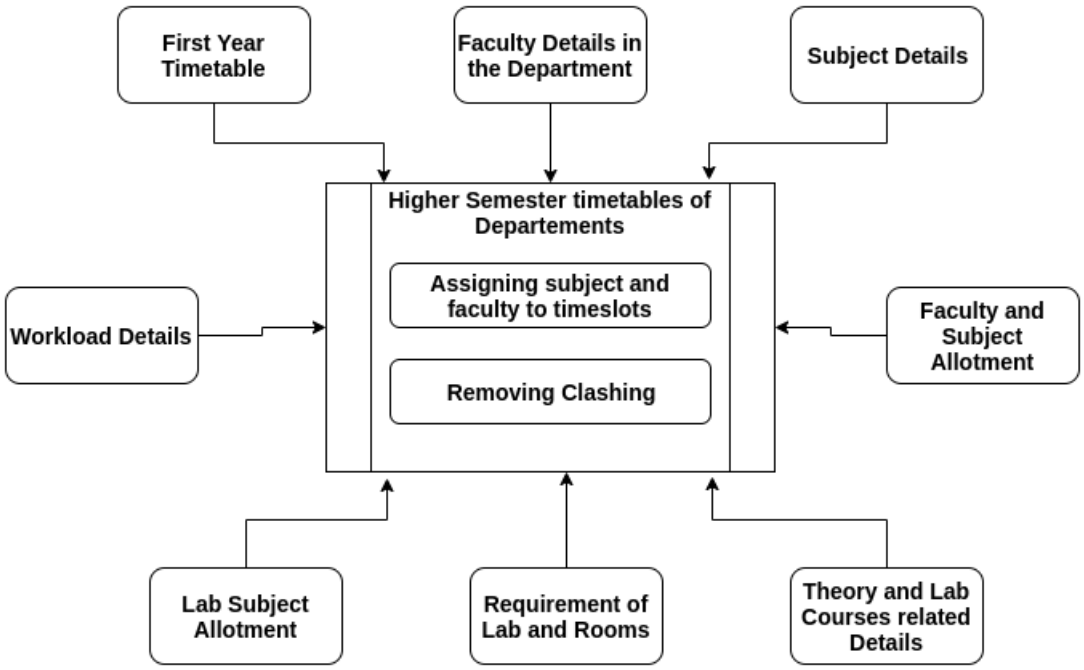


Figure 3.1: The Proposed System/System Architecture

Design of a scheme is basically a blueprint or a plan of a system . It determines the parts/ structure of components. Design contains the following things:

- Faculty details in department tells that the details of respective faculty in department.
- A workload detail tells that the higher and lower workload that faculty has based on their designation. That is for professor has less work than the assistant professor.
- Subject details as subject name, subject code.
- Faculty and subject allotment table consist for which subject respective faculty is allotted based on timeslots.
- Theory and lab courses related details contain the details of each subject that is handled by respective faculty.

3.3.1 Detailed Design

Use Case :-

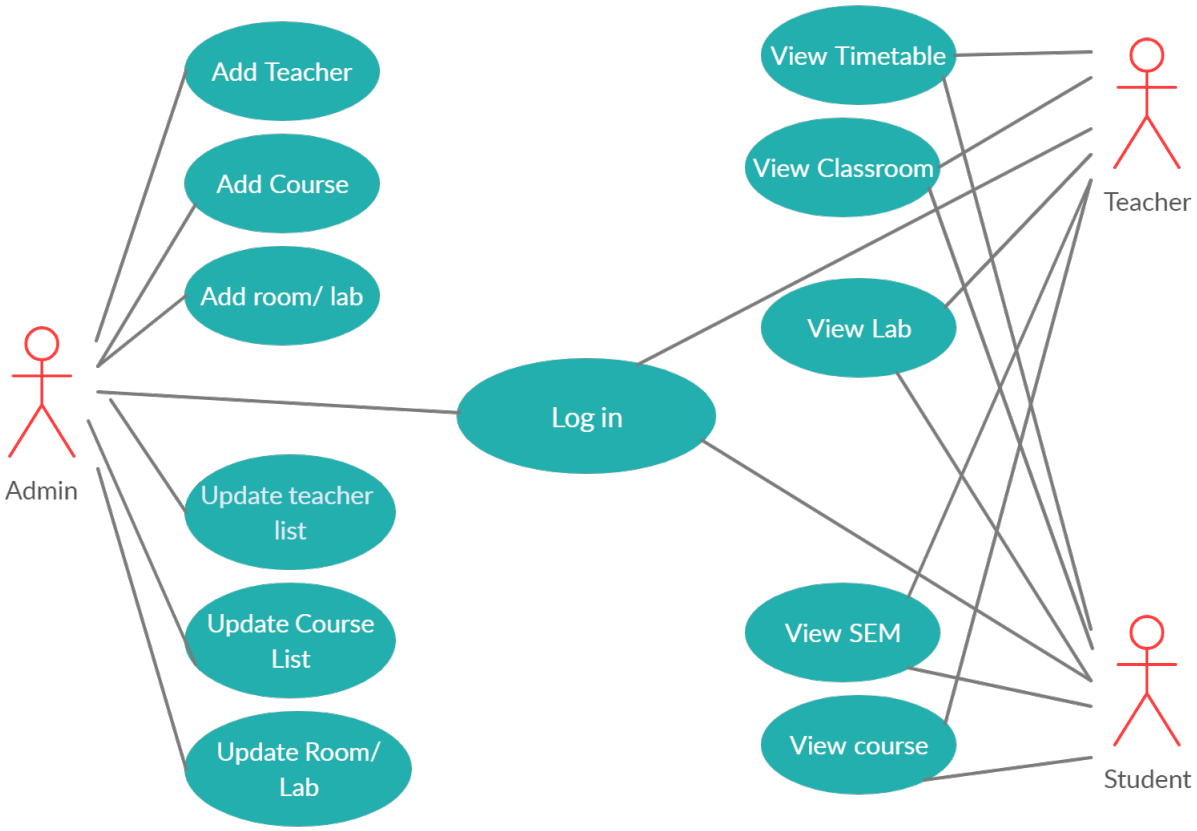


Figure 3.2: Use Case

The Use Case diagram is a graphical depiction of the interactions among the elements of Timetable Management System. It represents the methodology used in system analysis to identify,clarify and organize system requirements of Timetable Management System in this Use Case Diagram are: Admin, teacher, Student who perform different type of use cases such as Timetable,Manage Class,Manage Teacher,Manage Subjects and Full Timetable Management. Major elements of UML Use case Diagram of Timetable Management System is shown in the Picture Above.

Activity Diagram :-

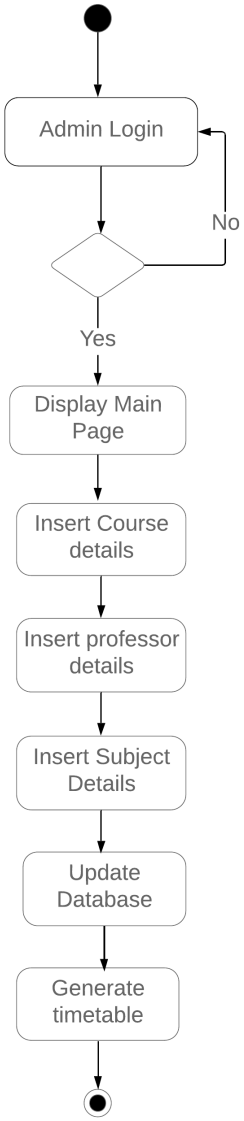


Figure 3.3: Activity Diagram

This is the login activity diagram of Timetable Management System , which shows the flows of Login Activity,Where admin will be able to login using their username and password. After login user can manage all the operation on Attendence,Subject,Exam,Student,Class. All the Pages such as Student Class are secure and user can access these page after login. The diagram below helps demonstrate how the Login Page works in a Timetable Management System The Various object in the Student,Subject and class , and user will not be able to access this page without verifying their identity.

Login Sequence Diagram :-

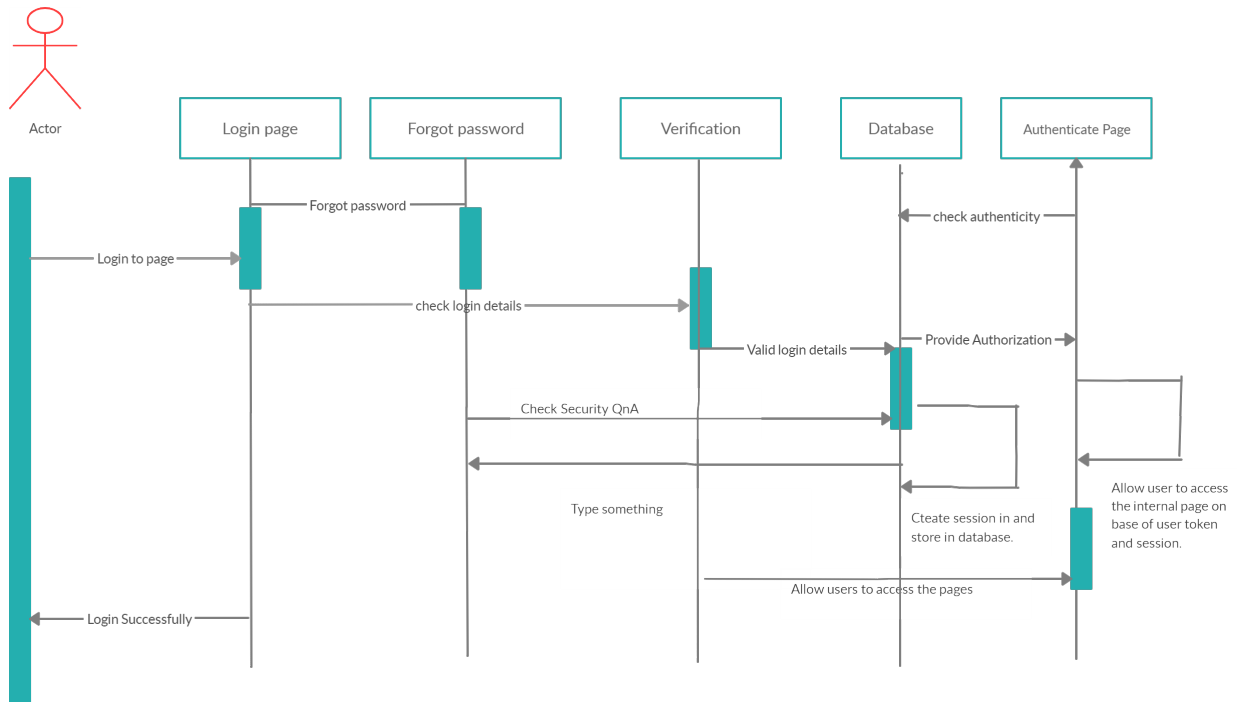


Figure 3.4: Login Sequence Diagram

This is the Login Sequence Diagram of Timetable Management System, where admin will be able to login in their account using their credentials, After login user can manage all the operations on Teacher, Class/Room, Teacher. All the pages such as Teacher, Class, Timetable are secure and user can access these page after login. The Diagram above helps demonstrate how the login page works in a Timetable Management System.The various objects in the Class Room/Lab, Teacher, Courses and Timetable Page-interact over the course of the sequence and user will not be able to access this page without verifying their identity.

Sequence Diagram for Timetable Management :-

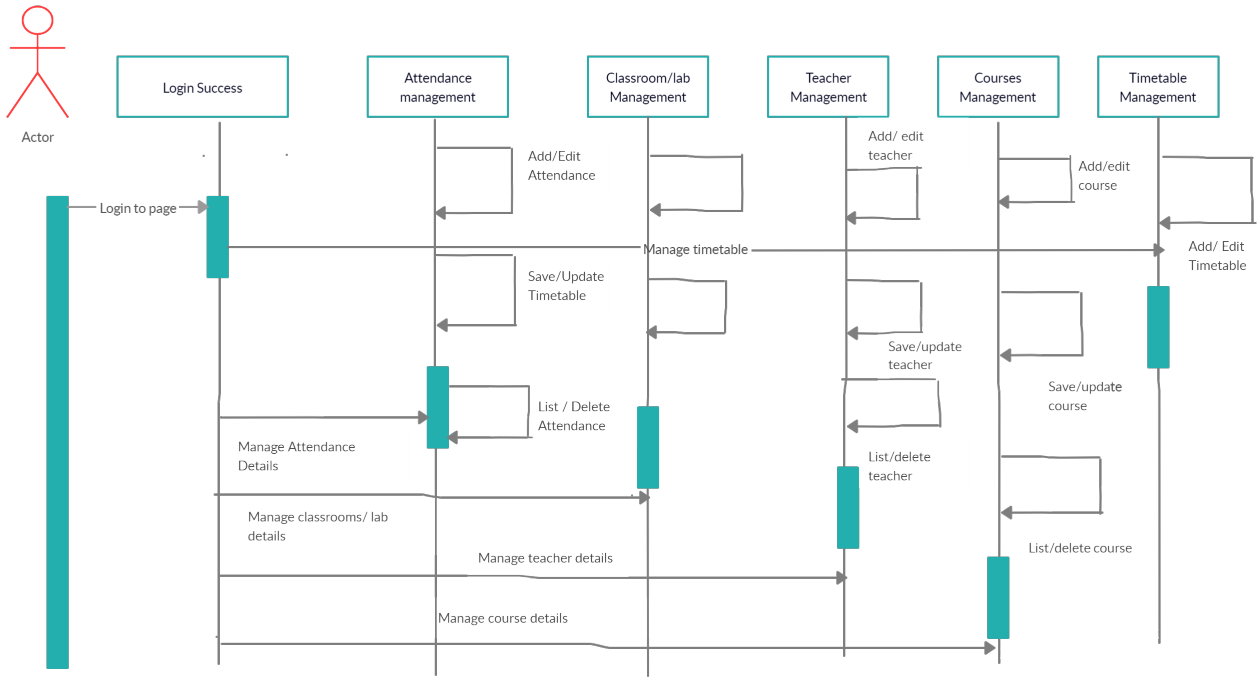


Figure 3.5: Sequence Diagram for Timetable Management

This is UML sequence diagram of Timetable Management System which shows the interaction between the objects of Teacher, Class/Room, Courses, Timetable. The instances of class objects involved in this UML Sequence Diagram of Timetable Management System are as follows:

- Teacher Object
- Class Room/Lab Object
- Course Object
- Timetable Object

3.4 Methodology/Procedures

There are two ways or two algorithms that can used to implement this Project and they are:

- Genetic Algorithm
- Ant Colony Optimization

Genetic Algorithm :-

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

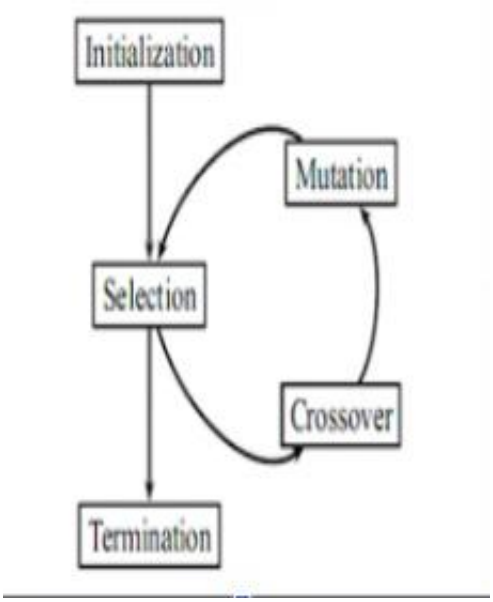


Figure 3.6: Flow For Genetic Algorithm

We are going develop the project using Genetic Algorithm as the GA works for both strong and weak constraints.

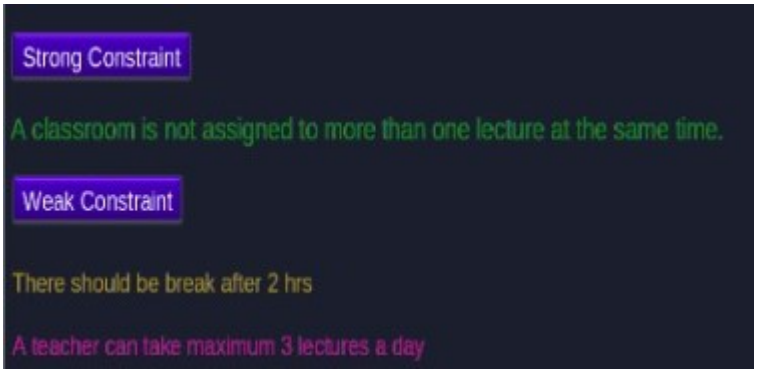


Figure 3.7: Strong and Weak Constraints

3.4.1 Mathematical Formula

$$\textbf{Fitness} = 1 \;/\; ((1.0 * \textbf{Number of conflict} + 1))$$

Chapter 4

Results & Discussions

4.1 Implemented Algorithm's Pseudo Code

Algorithm for Genetic:

- **Step 1: Initialize the Data Set**

  Example :-

```
rooms = ["R1","R2","R3","R4",]

meeting_times = 35

instructors = [["I1","Dr James Web"],
               ["I2","Mr Mike Brown"],
               ["I3","Dr Steve Day"],
               ["I4","Mrs Jane Doe"],
               ["I5","Mr Bhaven Gore"],
               ["I6","Mrs Bhavika Gore"]]

courses = [["C1","325K",[instructors[0]]],
           ["C2","319K",[instructors[1]]],
           ["C3","462k",[instructors[2]]],
           ["C4","464K",[instructors[4],instructors[5]]],
           ["C5","360C",[instructors[3]]],
           ["C6","303K",[instructors[4],instructors[0]]],
           ["C7","303L",[instructors[5]]]]

depts = [["SEA", [courses[0], courses[2]]],
         ["SEB" , [courses[1], courses[3], courses[4]]],
         ["BE", [courses[5], courses[6]]]]
```

Figure 4.1: Initialization

- **Step 2: Create Population**

  This will take these dataset and make a Schedule and a group of 9 Schedule is termed as a Population.

  Example :- Here we take 9 set of schedule for your simplicity.

[**[Schedule 1]**,
[Schedule 2],
[Schedule 3],
[Schedule 4],
[Schedule 5],
[Schedule 6],
[Schedule 7],
[Schedule 8],
[Schedule 9]]

Each Schedule Consists of 70 Classes

[Schedule] —->>**[[Class 1],[Class 2],[Class 3].....[Class 70]]**

Each Class Consists of 3 parameters

[Class]———>>[Course Name , Room Number , Instructor List]

- **Step 3: Calculate Fitness of the population**

  We then take this whole population and Calculate fitness by taking one Schedule at a time.

  Fitness Calculation

  In our Project

  Fitness =1 / ((1.0*Number of conflict + 1))

  The Conflicts are :-

  - No Two Classes should have the same Teacher at the same time.
  - No Two Lectures should take place at the same room at the same time.
  - Number of hours of a particular lec should be maintained.

  If any of the above conflict occur then number of conflict parameter increases.

  After Calculating fitness we sort it

  If Fitness is Equal to 1 i.e Number of Conflicts is Equal to 0

  Then,

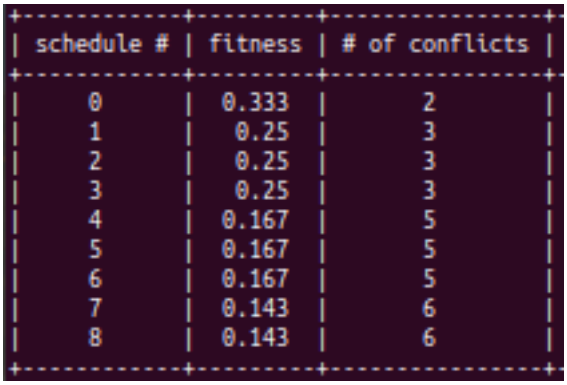
  Print the Table Generated

  Else

  - **3.1 Crossover the Population**

    In Crossover we take 2 schedules at a time from the population and perform crossover operation.

    Example :-



schedule #	fitness	# of conflicts
0	0.333	2
1	0.25	3
2	0.25	3
3	0.25	3
4	0.167	5
5	0.167	5
6	0.167	5
7	0.143	6
8	0.143	6

Figure 4.2: Fitness

Schedule 1 —->[[Class 1],[Class 2],[Class 3].....[Class 70]]
Schedule 2 —->[[Class 1],[Class 2],[Class 3].....[Class 70]]

We then set crossover rate at **0.5**
Using this we form a new schedule
We generate a random number and compare it with the crossover rate.
For all 70 Classes

If rand >rate then, new Schedule of that class —>Schedule 1 of that class
Else new Schedule of that class —->Schedule 2 of that class.

- **3.2 Mutate the Crossover Population**
  In mutation we take the crossover Schedules one at a time and perform Mutation
  Example :-

  First we Initialize a new Schedule then we take the crossover Schedule

  Crossover Schedule 1 —- >[[Class1] , [Class2] , [Class3] ..... [Class 70]]

  We set Mutation rate at **0.01**
  We generate a random number and compare it with the crossover rate.
  For all 70 classes

  If rand >0.01 then,
  Crossover Schedule of that class —>new Schedule of that class

- **3.3 Thus after Mutation we got the new Evolved Population.**

- **Step 4: Go to step 3**
- **Step 5: END**

4.2 Results

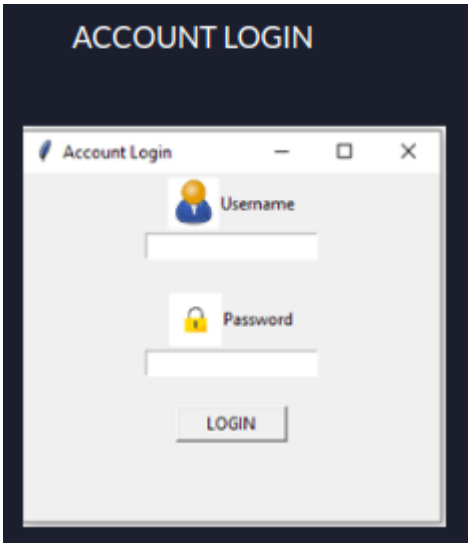


Figure 4.3: Account Login

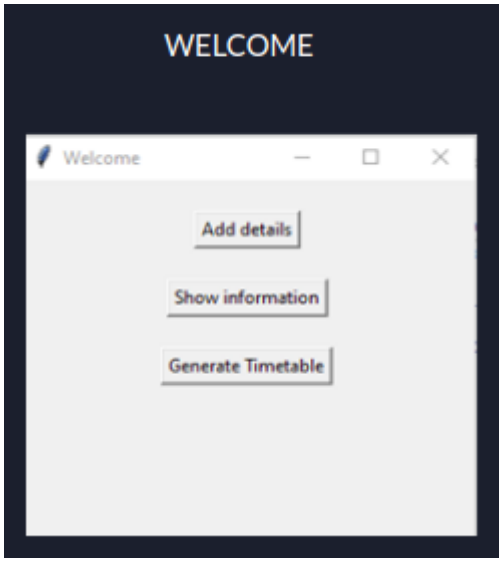
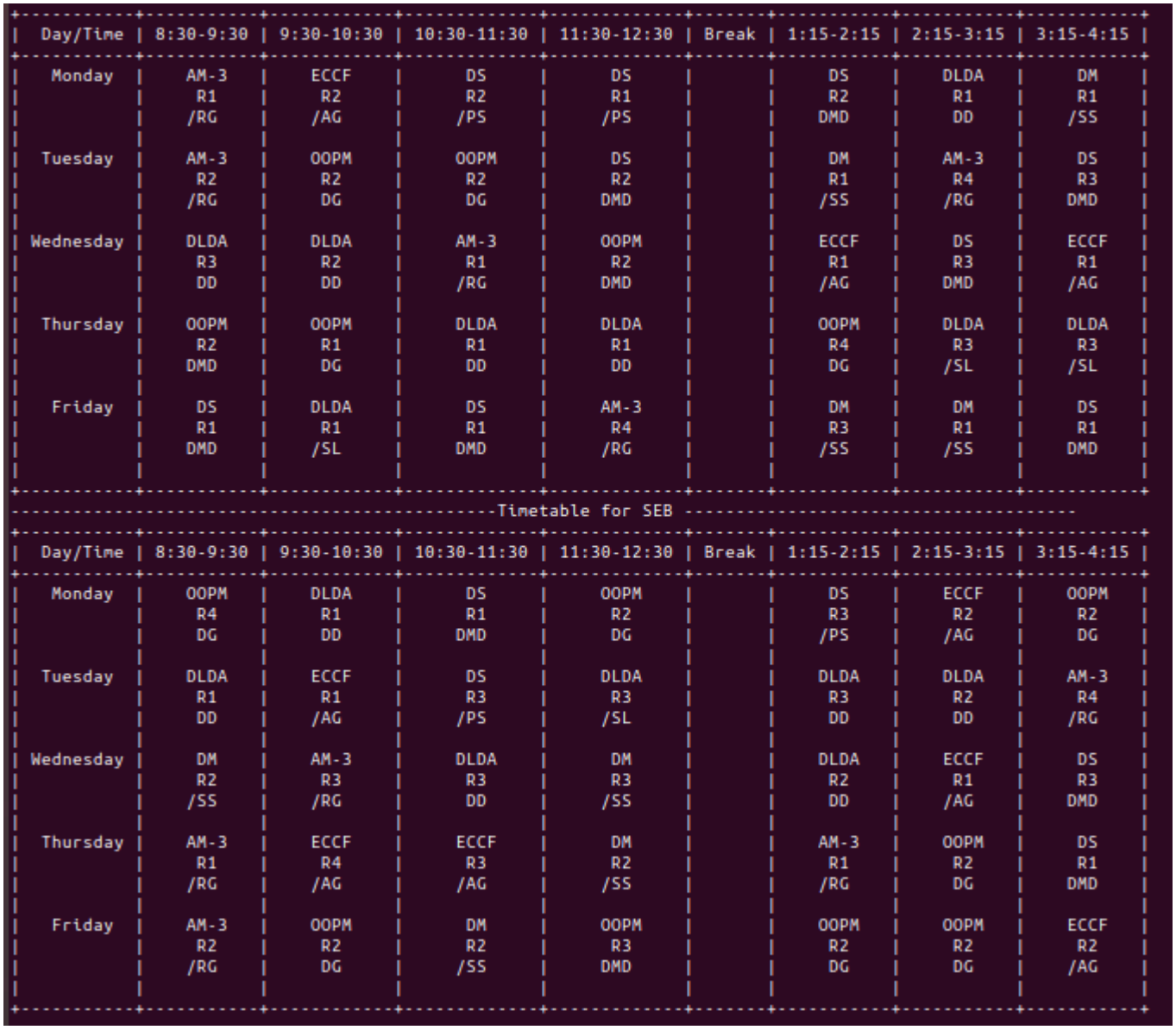


Figure 4.4: Welcome Page



Day/Time	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	Break	1:15-2:15	2:15-3:15	3:15-4:15
Monday	AM-3 R1 /RG	ECCF R2 /AG	DS R2 /PS	DS R1 /PS		DS R2 DMD	DLDA R1 DD	DM R1 /SS
Tuesday	AM-3 R2 /RG	OOPM R2 DG	OOPM R2 DG	DS R2 DMD		DM R1 /SS	AM-3 R4 /RG	DS R3 DMD
Wednesday	DLDA R3 DD	DLDA R2 DD	AM-3 R1 /RG	OOPM R2 DMD		ECCF R1 /AG	DS R3 DMD	ECCF R1 /AG
Thursday	OOPM R2 DMD	OOPM R1 DG	DLDA R1 DD	DLDA R1 DD		OOPM R4 DG	DLDA R3 /SL	DLDA R3 /SL
Friday	DS R1 DMD	DLDA R1 /SL	DS R1 DMD	AM-3 R4 /RG		DM R3 /SS	DM R1 /SS	DS R1 DMD

Timetable for SEB

Day/Time	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	Break	1:15-2:15	2:15-3:15	3:15-4:15
Monday	OOPM R4 DG	DLDA R1 DD	DS R1 DMD	OOPM R2 DG		DS R3 /PS	ECCF R2 /AG	OOPM R2 DG
Tuesday	DLDA R1 DD	ECCF R1 /AG	DS R3 /PS	DLDA R3 /SL		DLDA R3 DD	DLDA R2 DD	AM-3 R4 /RG
Wednesday	DM R2 /SS	AM-3 R3 /RG	DLDA R3 DD	DM R3 /SS		DLDA R2 DD	ECCF R1 /AG	DS R3 DMD
Thursday	AM-3 R1 /RG	ECCF R4 /AG	ECCF R3 /AG	DM R2 /SS		AM-3 R1 /RG	OOPM R2 DG	DS R1 DMD
Friday	AM-3 R2 /RG	OOPM R2 DG	DM R2 /SS	OOPM R3 DMD		OOPM R2 DG	OOPM R2 DG	ECCF R2 /AG

Figure 4.5: Master Timetable

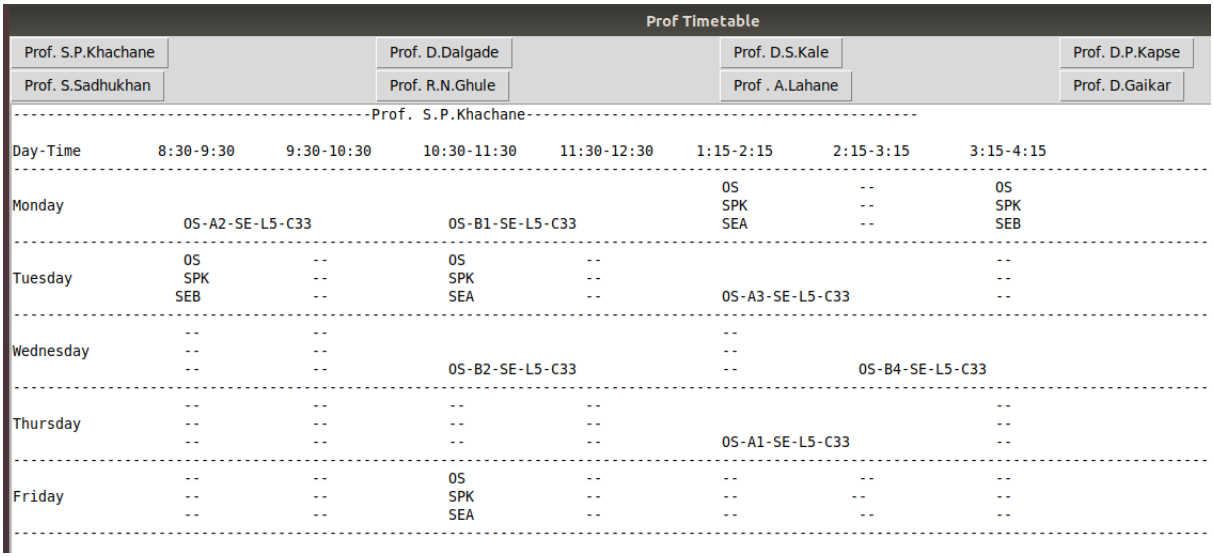


Figure 4.6: Faculty Timetable

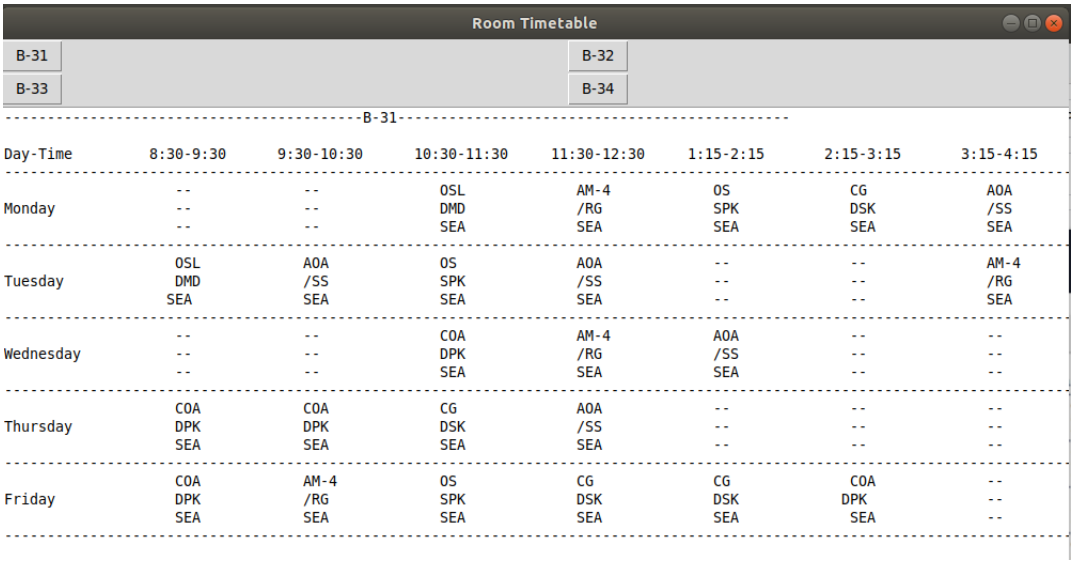


Figure 4.7: Room Timetable

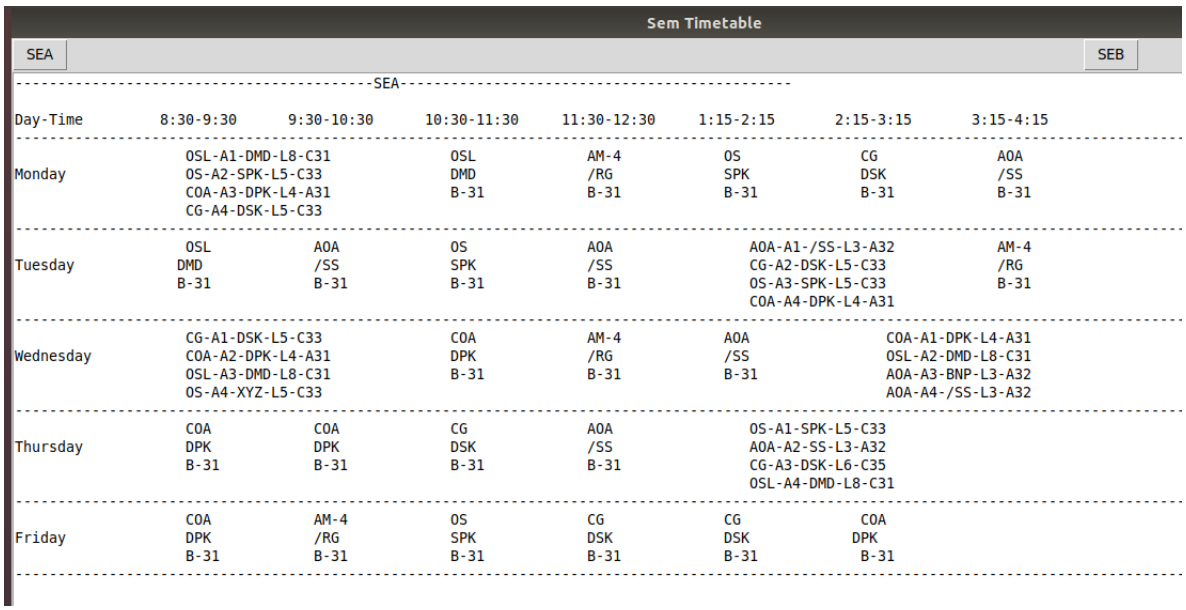


Figure 4.8: Class Timetable

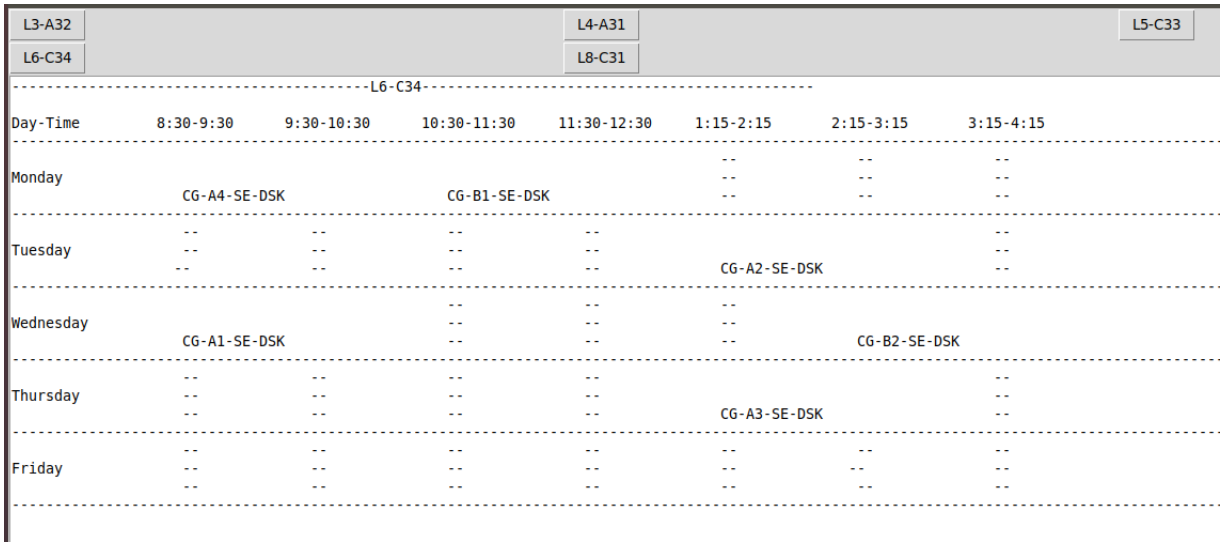


Figure 4.9: Lab Timetable

Chapter 5

Conclusion

It is complicated task that to handle many Faculty's and allocating subjects for them at a time physically. So our proposed system will help to overcome this disadvantage. Thus we can produce timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools are widely applicable and free to use also.

Bibliography

[1] Dipti Srinivasan Tian Hou Seow Jian Xin Xu, *Automated timetable generation using multiple context reasoning for university models*, 2002 IEEE conference.

[2] Anuja Chowdhary,Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, Dinesh Gawande *TIMETABLE GENERATION SYSTEM*, A paper published in IJCSMC Vol. 3, Issue. 2, February 2014.

[3] Saritha M, 2. Pranav Kiran Vaze, 3. Pradeep, 4. Mahesh N R (1.Assistant Professor), (2, 3, 4 UG Scholar), (1, 2, 3, 4 Department of CSE, SDMIT Ujire, Karnataka, India),*Automatic Time Table Generator* , Volume 7, Issue 5, May 2017 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

[4] S. Abdullah, E. K. Burke and B. McCollum ,' *A Hybrid Evolutionary Approach to the University Course Timetabling Problem*, Proceedings of the IEEE Congress Evolutionary Computation, Singapore,(2007).

[5] Lahoti, Y., Aaditya Punekar, H. P. (2015),' *Automated Timetable Generator* , International Journal of Science and Research, 4 .

[6] https://www.researchgate.net/publication/326265336 A STUDY ON AUTOMATIC TIMETABLE GENERATOR .

[7] https://medium.com/@vijinimallawaarachchi/time-table-scheduling-2207ca593b4d

Publication

Paper entitled "Institute Timetable Scheduler" is yet to be presented at *International Conference on Artificial Intelligence, Robots and Mechanical Engineering(ICAIRME - 2020)*

The presentation of the paper at the Conference is delayed due to the COVID-19 issue, we are waiting for further response.

Annexure

=210mm =297mm

Institute Timetable Scheduler

[1]Prof. Sunil P Khachane , [2]Bhaven Gore , [3]Disha Shirdhankar , [4]Giriraj Belanekar

[1]Professor , [2,3,4]Students, MCT Rajiv Gandhi Institute of Technology, Off Juhu Versova Link rd, Versova, Andheri (W), Mumbai-53

Email: [1]sunil.khachane@mctrgit.ac.in , [2]gorebhaven7@gmail.com , [3]disha.shirdhan@gmail.com , [4]girirajbelanekar8@gmail.com

Contact: [1]9833800580 , [2]9867323655 , [3]9082627804 ,[4]9137036912

Abstract: This paper presents a better approach to prepare the timetable of the university rather than using the manual system. The manual system of creating a timetable for the college with a large number of students is very time consuming and usually ends up with various classes clashing either at the same room or with the same teacher having more than one class at a time. To overcome these hectic problems we propose to make an automated scheduler. The system we will develop will take various inputs like details of students, subjects and classrooms depending upon these inputs it will create or generate a possible timetable. Making best utilization of all resources in a way that will best opt any of constraints or college rules. This can be achieved by using Genetic Algorithm.

Index terms: Scheduler, Timetable, Genetic Algorithm.

I. INTRODUCTION

A manual lecture time-table scheduling demands considerable time and efforts. Although, most of the work done in the administrative field has been digitised, the lecture time table scheduling is still done manually across universities due to its inherent difficulties.

This scheduling problem is based on constraint satisfaction; in which we intend to find a solution which satisfies the given set of constraints. This problem draws its roots to the 'NP-Hard' class of problems in which the computational time required for scheduling tends to grow exponentially as the number of variables increases. Through our project, we reckon a realistic time-table algorithm which is capable of handling hard as well as soft constraints.

.

II. LIMITATIONS OF EXISTING SYSTEMS

All the Existing methods used were either manual, local search or using algorithms whose time complexity reaches polynomial time.

As mentioned when Timetable generation is done, it should consider the maximum and minimum workload that is in a college. In those cases, timetable generation will become more complex. Also, it is a time consuming process.

We know all institutions or organisations have their own timetable; managing and maintaining these will not be difficult. Considering workload with this scheduling will make it more complex.

III. PROPOSED SYSTEM

Our paper entitled "Institute Timetable scheduler" is meant to generate timetable scheduling processes in colleges or in any other institutions which could minimize the human effort and maximise the efficiency and the timetable is stored in excel sheets.

Main Objectives are :-

- The final system should be able to generate time tables in a completely automated way which will save a lot of time and effort of institute administration.
- To make the timetable system generic so that it can work equally well for different schools, colleges and universities.
- User defined constraints handling.
- Ease of use for users of the system so that he/she can make automatic timetable..
- Focus on optimization of resources i.e teachers, labs and rooms etc
- Provide a facility for everyone to view the timetable.

- Generate multiple useful views from timetab

=210mm =297mm

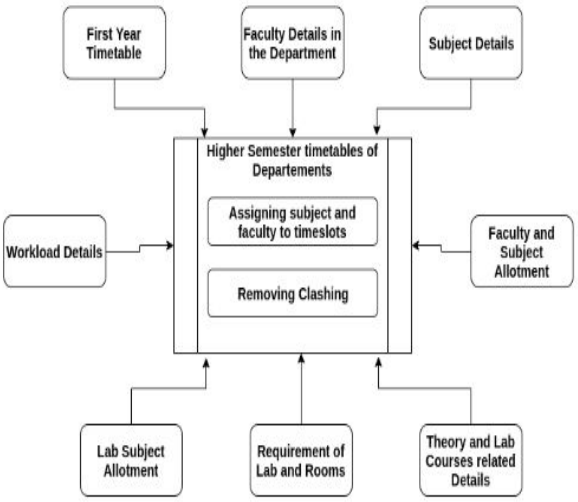


Figure 1: Detailed Design

IV. METHODOLOGY / PROCEDURES

Genetic Algorithm -

Genetic Algorithm (GA) is an adaptive heuristic search algorithm that belongs to the larger part of evolutionary algorithms.Genetic algorithm is based on the ideas of natural selection and genetics. It is intelligent in exploitation of random search provided with historical data to direct the search into the region of better performance in solution space.

It is commonly used to generate high-quality solutions for optimization problems and search problems.

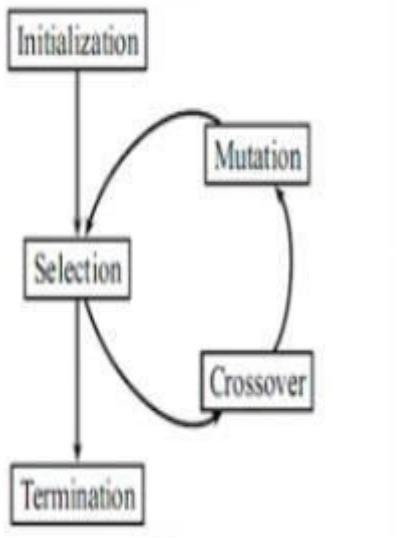


Figure 2: Flow of Genetic Algorithm

Procedure

Algorithm for Genetic:

Step 1: Initialize the Data Set

Example :-

```
rooms = ["R1","R2","R3","R4",]

meeting_times = 35

instructors = [["I1","Dr James Web"],
               ["I2","Mr Mike Brown"],
               ["I3","Dr Steve Day"],
               ["I4","Mrs Jane Doe"],
               ["I5","Mr Bhaven Gore"],
               ["I6","Mrs Bhavika Gore"]]

courses = [["C1","325K",[instructors[0]]],
           ["C2","319K",[instructors[1]]],
           ["C3","462k",[instructors[2]]],
           ["C4","464K",[instructors[4],instructors[5]]],
           ["C5","360C",[instructors[3]]],
           ["C6","303K",[instructors[4],instructors[0]]],
           ["C7","303L",[instructors[5]]]]

depts = [["SEA", [courses[0], courses[2]]],
         ["SEB" , [courses[1], courses[3], courses[4]]],
         ["BE", [courses[5], courses[6]]]]
```

Figure 3: Initializing Data

Step 2: Create Population

This will take these dataset and make a Schedule and a group of 9 Schedule is termed as a Population.

Example :- Here we take 5 sets of schedules for your simplicity.

[[Schedule 1],

[Schedule 2],

[Schedule 3],

[Schedule 4],

[Schedule 5]]

Each Schedule Consists of 70 Classes

[Schedule] >> [[Class 1],[Class 2],[Class 3]..... .....[Class 70]]

Each Class Consists of 3 parameters

[Class] >> [Course Name , Room Number , Instructor List]

Step 3: Calculate Fitness of the population

We then take this whole population and Calculate fitness by taking one Schedule at a time.

Fitness Calculation

Fitness =1 / ((1.0*Number of conflict + 1))

The Conflicts are :-

– No Two Classes should have the same Teacher at the same time.

– No Two Lectures should take place at the same room at the same time.

– Number of hours of a particular lec should be maintained.

If any of the above conflicts occur then the number of conflict parameters increases.

After Calculating fitness we sort it

schedule #	fitness	# of conflicts
0	0.333	2
1	0.25	3
2	0.25	3
3	0.25	3
4	0.167	5
5	0.167	5
6	0.167	5
7	0.143	6
8	0.143	6

Figure 4: Fitness

If Fitness is Equal to 1 i.e Number of Conflicts is Equal to 0

Then,

Print the Table Generated

Else

– 3.1 Crossover the Population

In Crossover we take 2 schedules at a time from the population and perform crossover operation.

Example :-

Schedule 1 > [[Class 1],[Class 2],[Class 3] ... .[Class 70]]

Schedule 2 > [[Class 1],[Class 2],[Class 3]..... .....[Class 70]]

We then set crossover rate at **0.5**

Using this we form a new schedule

We generate a random number and compare it with the crossover rate.

For all 70 Classes

If rand > rate then,

new Schedule of that class > Schedule 1 of that class

Else

new Schedule of that class > Schedule 2 of that class.

– 3.2 Mutate the Crossover Population

In mutation we take the crossover Schedules one at a time and perform Mutation

Example :-

First we Initialize a new Schedule then we take the crossover Schedule

Crossover Schedule 1 > [[Class1] , [Class2] , [Class3] . . . . . [Class 70]]

We set Mutation rate at **0.01**

We generate a random number and compare it with the crossover rate.

For all 70 classes

If rand > 0.01 then,

Crossover Schedule of that class > new Schedule of that class

– 3.3 Thus after Mutation we got the new Evolved Population.

- **Step 4: Go to step 3**
- **Step 5: END**

V. SIMULATIONS AND EXPERIMENTAL RESULTS

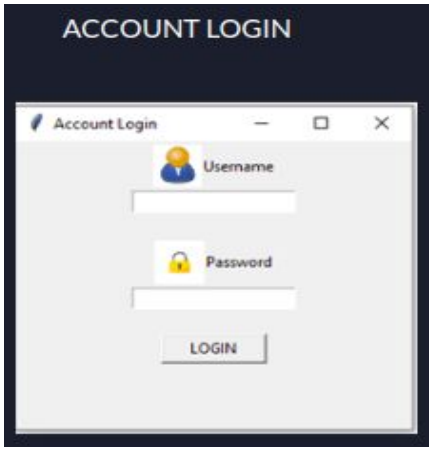


Figure 5: Account Login

Figure 6: Welcome Page

Figure 7: Add Details Page

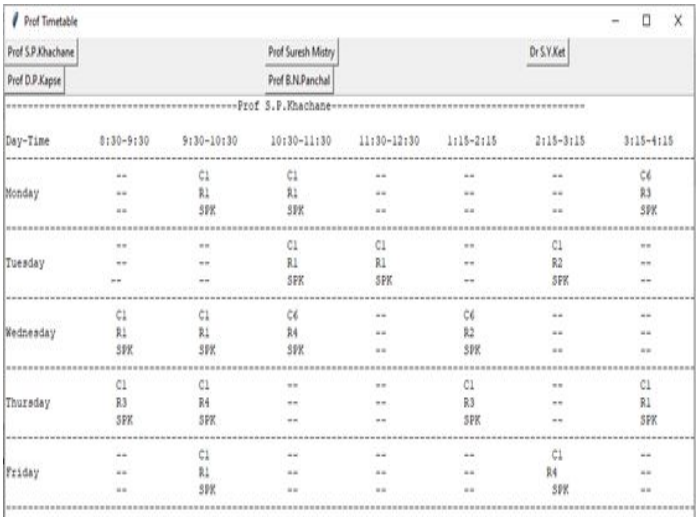


Figure 8: Faculty Timetable

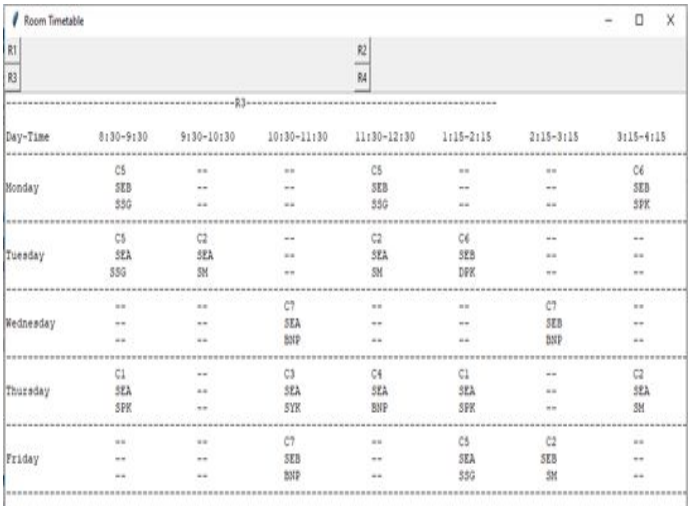


Figure 9: Room Timetable

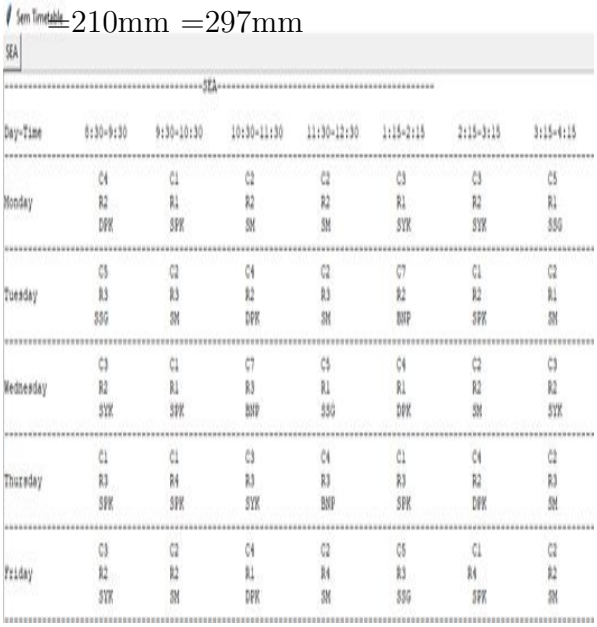


Figure 10: Class Timetable

CONCLUSION

It is an intricate task to organize and manage the schedule of the entire faculty throughout the university and also assign specific subjects to them as per their expertise which fulfills the requirement of the students. Our proposed system will help to overcome the impediment faced because of physical means. Our system has the potential to generate a time-table for any number of courses across departments/sections in a university In the broader aspect, this can also be put to use by various commercial organizations too.

REFERENCES

[1] Dipti Srinivasan, Tian Hou Seow, Jian Xin Xu,Automated timetable generation using multiple context reasoning for university models, 2002 IEEE conference.

[2] Anuja Chowdhary, Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, Dinesh Gawande TIMETABLE GENERATION SYSTEM, IJCSMC Vol. 3, Issue. 2, February 2014.

[3] Saritha M, Pranav Kiran Vaze, Pradeep, Mahesh N R Automatic Time Table Generator, Volume 7, Issue 5, May2017 ISSN: 2277 128X International Journal of Advanced Research in ComputerScience and Software Engineering.

[4] S. Abdullah, E. K. Burke and B. McCollum,'A Hybrid Evolutionary Approach to the University Course Timetabling Problem,Proceedings of the IEEE Congress Evolutionary Computation, Singapore,(2007).

[5] Lahoti, Y., Aaditya Punekar, H. P. (2015),'Automated Timetable Generator, International Journal of Science and Research, 4.

[6] https://www.researchgate.net/publication/326265336 A STUDY ON AUTO-MATIC TIMETABLE GENERATOR.

★★★

Acknowledgement

We wish to express our sincere gratitude to Dr. Sanjay U. Bokade, Principal and Dr. Satish. Y. Ket , H.O.D of Department Computer Engineering of Rajiv Gandhi Institute of Technology for providing us an opportunity to do our project work on "Institute Timetable Scheduler".

This project bears on imprint of many peoples. We sincerely thank our project guide Prof. Sunil.P.Khachane for his guidance and encouragement in carrying out this synopsis work.

Finally, we would like to thank our colleagues and friends who helped us in completing project work successfully

1. Giriraj Belanekar
2. Bhaven Gore
3. Disha Shirdhankar