

# Survey Paper on Automatic Timetable Generator

Ankit Pounikar<sup>1</sup>, Hrushikesh Bhandage<sup>2</sup>, Nupur Dalvi<sup>3</sup>, Tanvi Borade<sup>4</sup>, S. H. Lokhande<sup>5</sup>

Students, Department of Computer Engineering<sup>1,2,3,4</sup>

Assistant Professor, Department of Computer Engineering<sup>5</sup>

Sinhgad Institute of Technology and Science, Pune, Maharashtra, India

**Abstract:** In today's literate world it is very difficult to create time table manually. Timetables are to be created uniquely for all branches and years respectively. It becomes a very hectic, time consuming, and needs manpower for preparing the timetables manually. In some cases, this process becomes complex when any staff is on a leave or needs to be substituted. It would be convenient if an algorithm creates timetable which will save a lot of time and reduce the load and pressure on the person doing the job. Using software to do the job saves a lot of time and can also create timetables for complex situations. It will also avoid any human error like: subject clash, vacant slots.

**Keywords:** Genetic Algorithm, Automated Timetable, soft and hard constrains

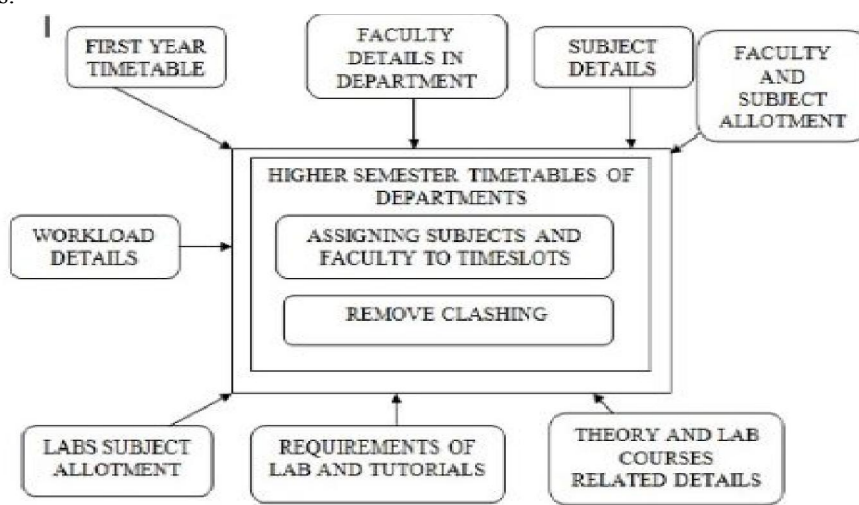
## I. INTRODUCTION

Lecture timetabling is a very important process and one of the common scheduling problems in any educational institution, which can be described as the allocation of resources for tasks under predefined constraints so that it maximizes the possibility of allocation or minimizes the violation of constraints.

A key factor in running an educational centre or basically an academic environment is the need for a well-planned, well-throughout and clash-free timetable. Back in the days when technology was not in wide use, the timetable was manually created by the academic institution.

## II. SYSTEM ARCHITECTURE

In previous chapter, an idea about how the project is started after studying various previous works and the techniques that will be used to implement the project are discussed. The review gives a brief idea about the papers considered and analysed & the summary gives the brief description and the observations regarding the same. This chapter includes the software requirements specifications in detail including the functional & non-functional requirements along with the system requirements.



**Figure 1:** System Architecture

The block diagram in Fig. 1 gives an overview of the approach towards building a basic version of the intended features for Timetable.



The workflow for timetable generation works in following manner –

- Admin will modify all the details of the student, faculty and subjects.
- Admin will generate the timetable by providing the input as subject, faculty, type, etc.
- Admin will update the timetable and notify the student and faculty.
- The timetable without any clashes and satisfying all the constraints will be generated.
- Appropriate lab or class will also be allotted for the session.
- Student and faculty will get the updated timetable message on their registered mobile number.
- Student and faculty can login their account using their personal login.
- The timetable can be viewed to the student as per their details provided.

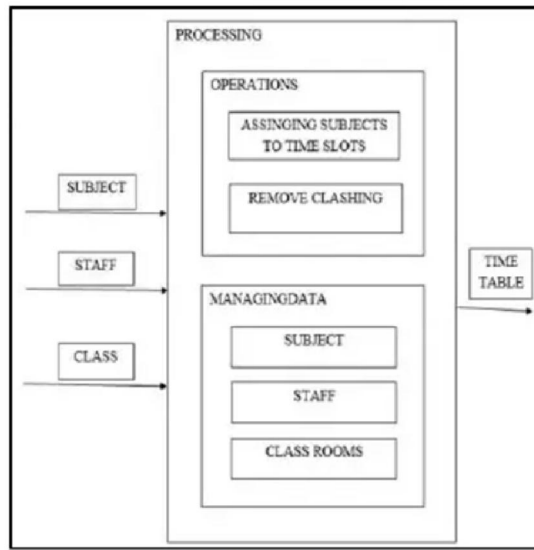


Figure 2: System Architecture

### 2.2 Data Flow Diagrams

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself.

#### 2.2.1 DFD (Level 0)

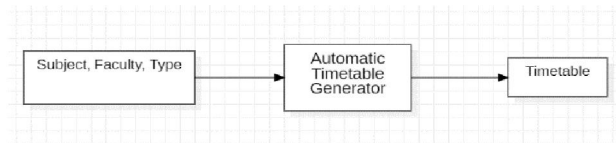


Figure 3: Level 0 Data Flow Diagram for Automatic Timetable Generator.

It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

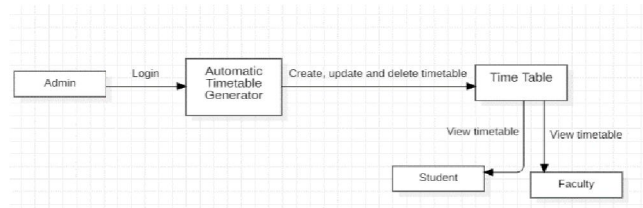
In figure 3 Subject, Faculty and Type of session is fed as input to the system and a Timetable is generated as output.

#### 2.2.2. DFD (Level 1)

In level-1 DFD, context diagram is decomposed into multiple bubbles/processes.

#### 2.2.3. DFD (Level-2)

Level-2 DFD goes one step deeper into parts of 1-level DFD. It is used to plan or record the specific/necessary detail about the system's functioning. In figure 4 the detailed process extended from DFD (Level-1) are shown.



**Figure 4:** Level 1 Data Flow Diagram for Automatic Timetable Generator

### III. LITERATURE SURVEY

In the paper {1}, the proposed algorithm can be further adapted to suit as per the requirements of different institutes and universities. It considers a wide range of constraints divided into soft and hard, to generate faculty, classroom/laboratory and student timetables. The complexity of the proposed algorithm is  $n^3$ .

In {2}, the paper presents a categorisation of the methodologies conducted in recent years based on chronology, category and application.

{3} We propose two pattern-based formulations and a solution algorithm that simultaneously exploits column generation and a team of metaheuristics to build and improve solutions.

{4} Graph Colouring Algorithm (GCA) takes 11 percent of the time, Genetic and Graph Colouring (GCA) takes 25 percent, Heuristic and Iterated Local Search Algorithms (ILSA) take 44 percent of time, and Heuristic Algorithm takes 20 percent.

{5}. This means developers can use imperative programming for business logic and declarative programming for user interface (UI) development.

{6}. The fitness score relates to the quantity of crashes the timetable has experienced.

#### 3.1. Literature Summary

The above survey of various researchers of different algorithms of Automated Time-Table Generator.

Trying to develop a software which helps to generate Timetable for an Institution automatically. By looking at the existing system we can understand that timetable generation is done manually.

Manually adjust the timetable when any of the faculty is absent, and this is the big challenge for Automatic Timetable Generator that managing the timetable automatically when any of the faculty is absent.

As mentioned, when Timetable generation is being done, it should consider the maximum and minimum workload that is in a college. Also, it is a time-consuming process.

### IV. SUMMARY

This report studied various papers. The highlights and observation are reported and the gap has been analysed, based on which problem statement is designed along with its objectives. This report addresses the procedure of the timetable generation which will be consistently easier with Automated System for Timetable Generation than using spreadsheet manually which might lead to constraints problem that are arduous to establish when timetable is generated manually. The Automated System for Timetable Generation is provided with necessary details of courses, classrooms and faculties which are stored in database (SQL server). The Automated System for Timetable Generation generates timetable based on these details in database with minimum time and satisfies all the constraints.

### REFERENCES

- [1]. Sara Ceschia, Andrea Schaerf, Educational timetabling: Problems, benchmarks, and state-of-the-art results (Survey), Submitted in 2021, accepted in 2022
- [2]. Mrunmayee V. Rane, Vikram M. Apte, Vishakha N. Nerkar Mani Roja Edinburgh, K. Y. Rajput "Automated Timetabling System for University Course" 5-7 March 2021
- [3]. Joo Siang Tan, Say Leng Goh, Graham Kendall, Nasser R. Sabard "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems" 1 March 2021

- [4]. Landir Saviniec, Maristela O. Santos, Alysson M.Costa, Lana M.R. dos Santos “Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems”1 Feb 2020
- [5]. Tiny Wijerathna Ekanayake, Pavani Subasinghe, Shawn Ragel, Anjalie Gamage, Suchini Attanayaka,” Intelligent Timetable Scheduler: A Comparison of Genetic Graph Coloring, Heuristic and Iterated Local Search Algorithms”, December 5-6,2019.
- [6]. Sundresan A/L Perumal, Mujahid Tabassum, Norita MD Norwani, Ganthan A/L Narayana Samy, Sivanathan A/L Perumal,” Development of an Efficient Timetable System using AngularJs and Bootstrap3” 2018
- [7]. Namrata Bodas, Jhanvi Shah, Yash Shah, Aishwarya Sontakke, Dnyaneshwar Dhangar. “Exam Cell Automation System and Timetable Generator” 2018
- [8]. Yash Lahoti, Aaditya Punekar, Hiten Patel, Vishal Bhimsariya,” Automated Timetable Generator”, Jan 2017