

ISSN 2063-5346



AUTOMATED TIME TABLE GENERATOR USING SPREAD SHEET FOR EDUCATIONAL MODULES

Neha Pradyumna Bora¹, Pradyumna Mulchand Bora²,
Dinesh Chandra Jain³

Article History: Received: 10.05.2023

Revised: 29.05.2023

Accepted: 09.06.2023

Abstract

The creation of timetables is a challenging work, but it is crucial in educational institutions. Conflicts arise while manually creating a schedule. Due to numerous hard and soft constraints, creating an effective timeline is a difficult task. Creating different schedules, such as the master timetable, class timetables, personal timetables, etc., takes a lot of time on the part of Timetable In-charge. Additionally, it takes time to distribute timetables to different parties. By using inputs like the number of subjects, teachers, workload of a teacher, semester, and priority of subjects, this automatic timetable generator software creates the timetables automatically. Timetable incharge must manually input the master timetable into the builder created in Google spreadsheet, which helps them detect various restrictions like resource conflicts, alerts of back-to-back lectures, etc.

This generator automatically creates all other timetables after adding the master schedule. This technology is actually used in one of the engineering college's 120 student departments. As a result, roughly 30+ timetables are automatically prepared. This builder saves an enormous time and helps the person in charge of creating the schedule by carefully taking all types of limits into account. Even the distribution of timetables may be done very easily by simply sharing them with all relevant parties and giving them the option to view them exclusively.

Keywords: Staff, Master Time Table, Class Time table, Individual Time table, Constraint

¹Assistant Professor, Department of Computer Engineering, SNJB's Late Sau KBJ CoE, Chandwad-423101, Maharashtra, India,

Email: muthaneha88@gmail.com

²Assistant Professor, Department of Mechanical Engineering, SNJB's Late Sau KBJ CoE, Chandwad-423101, Maharashtra, India,

Email: bora.pmcoe@snjb.org

³Professor, SAGE University, Indore, MP, India,

Email: dineshjain25210@gmail.com

DOI: 10.31838/ecb/2022.11.s3.765

1. INTRODUCTION

The timeline is the most crucial aspect of any institutional existence since it streamlines the entire institutional system. A timetable must take into account a variety of factors, including courses, faculties, classrooms, laboratories, timeslots, the number of students, restrictions, etc.

Creating a timetable for various events and activities is a complex task that requires careful consideration of multiple constraints and preferences. This paper presents an automated approach to generating timetables using Google Sheets, harnessing its powerful features for data management and manipulation. By leveraging the capabilities of Google Sheets, the proposed solution streamlines the timetable generation process, minimizing manual effort, and ensuring efficient allocation of resources. The system incorporates advanced functions and formulas, conditional formatting, and data validation techniques to produce accurate and flexible timetables. This paper provides a detailed overview of the methodology, discusses the implementation process, and highlights the advantages and limitations of using Google Sheets for automatic timetable generation.

The organization of this document is as follows. In Section 2 Literature Review is discussed. In Section 3 Design and methodology is described. Section 4 is related to results and upshots of the project and in the last conclusion is mentioned.

2. LITERATURE REVIEW

A large number of methods have been already proposed for solving timetabling problems [8] some of them are constraint-based methods, cluster methods, sequential methods, and meta-heuristic methods.

"A Genetic Algorithm Approach to University Timetabling" by E.K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem (1999) [1]:

This paper presents a genetic algorithm-based approach to university timetabling. The authors propose a chromosome representation for timetables and use fitness functions to evaluate the quality of generated solutions. The genetic algorithm optimizes the allocation of resources, such as classrooms and professors, while considering various constraints and preferences.

"A Survey on Timetabling Problems: A Classification, Benchmarking, and Analysis" by M.

Schaerf (1999) [2]:

This survey paper provides an extensive overview of different timetabling problems, including university course timetabling, school timetabling, and transportation timetabling. It discusses various techniques used for timetabling, such as constraint satisfaction, mathematical programming, and evolutionary algorithms. The paper also presents benchmark datasets and evaluates different algorithms' performance on these datasets.

"A Constraint Programming Approach to the Nurse Rostering Problem" by P. Prosser (1993) [3]:

Although specific to nurse rostering, this paper explores the application of constraint programming to timetabling problems. It discusses the use of constraint satisfaction techniques to generate high-quality timetables while considering a wide range of constraints, including shift preferences, nurse availability, and workload balancing.

"A Hybrid Genetic Algorithm for the Nurse Rostering Problem" by D. Abramson and S. Ashtiani (2007) [4]:

Focusing on nurse rostering, this paper presents a hybrid genetic algorithm that combines the genetic algorithm with local search methods. The authors demonstrate the effectiveness of their approach in generating feasible and high-quality nurse schedules. They incorporate various constraints, such as shift preferences, fairness, and continuity, into the genetic algorithm framework.

"An Effective Hybrid Algorithm for University Timetabling Problems" by L. Di Gaspero and A. Schaerf (2002)[5]:

This paper proposes a hybrid algorithm for university timetabling problems, combining constraint programming with tabu search. The authors define a model based on a constraint satisfaction framework and use tabu search to refine the solutions. The hybrid algorithm is evaluated on a set of benchmark instances, demonstrating its effectiveness in generating high-quality timetables.

"A Review of School Timetabling Problems and Solution Approaches" by E. K. Burke and B. McCollum (2007) [6]:

This review paper provides an overview of school timetabling problems and their solution approaches. It discusses various problem variants, including single and multi-period, teacher and classroom scheduling, and curriculum-based timetabling. The authors present different solution techniques, such

as graph coloring, constraint satisfaction, met heuristic algorithms, and evaluate their performance on benchmark instances.

By observing all the literature survey, we found Google Sheets is a web-based spreadsheet application offered by Google as part of its suite of productivity tools known as Google Workspace. It provides users with the ability to create, edit, and collaborate on spreadsheets online, making it a popular choice for individuals, businesses, and educational institutions. Google Sheets offers a wide range of features and functionalities that make it a powerful tool for data management, analysis, and visualization.

Advantages of Using Google Sheets for Timetabling:

2.1. Accessibility and Collaboration:

Google Sheets allows users to access and work on their timetables from anywhere with an internet connection. Multiple users can collaborate in real-time, making it ideal for teams or departments involved in the timetable generation process. Users can view changes, leave comments, and track revisions, ensuring seamless collaboration and communication.

2.2. Data Management and Manipulation:

Google Sheets provides a user-friendly interface for managing and manipulating data. It offers a variety of functions and formulas that enable automated calculations, making it easier to handle complex scheduling constraints and preferences. Users can sort, filter, and organize data efficiently, facilitating the process of generating timetables.

2.3. Advanced Functions and Formulas:

Google Sheets includes a wide range of built-in functions and formulas that can be leveraged for timetable generation. Functions like VLOOKUP, INDEX, MATCH, and IF statements allow for dynamic data retrieval and conditional calculations. These functions enable the implementation of rules and constraints, such as avoiding conflicts or optimizing resource allocation.

2.4. Conditional Formatting:

Conditional formatting in Google Sheets allows users to apply formatting rules based on specified conditions. This feature can be utilized to highlight conflicts, visualize scheduling constraints, or emphasize important events in the timetable. By applying different formatting styles and colors, users can easily interpret and analyze the generated timetable.

2.5. Data Validation:

Data validation ensures the accuracy and

consistency of input data. Google Sheets offers data validation options to define custom rules for cell values, including drop-down menus, numeric ranges, and text length restrictions. By implementing data validation, users can prevent errors and enforce adherence to specific constraints during the timetable generation process.

2.6. Integration with Other Google Services:

Google Sheets seamlessly integrates with other Google services, such as Google Calendar and Google Forms. This integration enables the synchronization of timetables with calendar events, making it convenient for individuals to view their schedules. Additionally, Google Forms can be used to collect data and preferences from participants, which can then be imported into Google Sheets for timetable generation.

3. DESIGN METHODOLOGY

Design Methodology of Time Table is based on following points:

1. Defining constraints and preferences for timetable creation.
2. Utilizing advanced functions and formulas for automated calculations.
3. Implementing conditional formatting for visual clarity.
4. Leveraging data validation techniques for error prevention.

Although resolving every soft constraint won't be necessary, doing so will be advantageous for all parties.

Constraints can be stated as:

- No more than one course may be enrolled in the same lab or classroom at once.
- It is not advisable to assign the same teacher to more than one course at once.
- There must be enough seats for the pupils in attendance.
- The teachers shouldn't be put in a time slot that they cannot handle.
- The daily maximum teaching hours for teachers should not be exceeded.
- The teacher should not provide back-to-back lectures; instead, he or she needs time to prepare.
- No batch should be left empty when scheduling practical classes so that every student is involved in their academic workload.
- Both rotations of a practical batch that take place within a week should be arranged in the same lab.

Modules:

1. Master timetable generation
2. Overlap detection and removal
3. Personal and laboratory timetable- template generation
4. Auto-updating timetable generation

Algorithm

1. Time table incharge uses the workload distribution chart and faculty limits as input to create the master schedule, taking into account things like list of subjects and their respective details (e.g., subject name, duration), list of available classrooms and their capacities, list of available teachers and their availability.
2. Generate a list of time slots for each day (e.g., morning, afternoon, evening) and prepare master schedule and other time table templates are created using formulas, by considering empty timetable grid with days as columns and time slots as rows.
3. In this system, conditional formatting is used to find overlaps.
For instance, if a professor has a lecture scheduled for 10.30 am on the second year of the class and has also been assigned a lecture for the fourth year at the same time, the cell will be marked in red to alert users to the overlap and prompt them to remove it right away.
4. Following timetables are generated automatically like Class Time table, Classroom Time table, Personal Time table, Lab Time table once master time table entry is completed.
5. In this software the creation of the master timetable is done manually with the help of automated sheets, and overlaps are found using conditional formatting rules, and templates are created using the aforementioned well-known functions such as index, and match.

4. RESULTS AND UPSHOTS

Google Spread Sheet Functions Used for creation on timetable

1. MATCH

The MATCH function in Google Sheets is used to search for a specified value within a range of cells and returns the relative position of that value in the range. It can be used for various purposes, such as finding the position of an item in a list, performing approximate matches, or creating dynamic lookups. MATCH(search_key, search_range, search_type)

2. INDEX

The INDEX function in Google Sheets is used to retrieve the value of a cell in a specific row and column within a range of cells. It is often used in combination with other functions like MATCH or VLOOKUP to perform more advanced lookups or data manipulations. INDEX(range, row_number, column_number);

3. IFERROR

The IFERROR function is used to handle errors that may occur in a formula or expression and return a specified value or perform an alternative action when an error is encountered. It allows you to control how errors are handled and display custom messages or values instead of error codes. IFERROR(value, value_if_error);

4. IMPORTRANGE

IMPORTRANGE is a function in Google Sheets that allows you to import data from one spreadsheet to another. It is a useful tool when you want to consolidate data from multiple sources or work with data from different spreadsheets in a single location.

Sample Code from Time Table Generation

=IFERROR(INDEX(MMT!\$C\$7:\$AA\$30,MATCH('226'!\$A\$5,MMT!\$G\$7:\$G\$30,2),''))

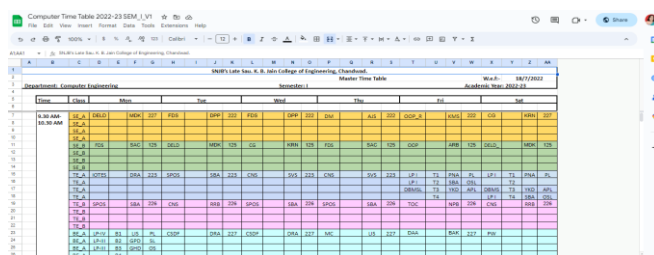


Figure 1. Snap of Master Time table

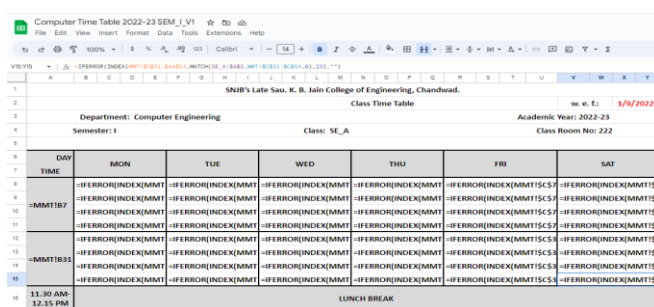


Figure 2. Snap of Class Time table with formula

DAY	MON	TUE	WED	THU	FRI	SAT
9.30 AM-10.30 AM	DELD MDK 227	FDS DPP 222	FDS DPP 222	DM AIS 222	OOP_R KMS 222	CG KRN 227
10.30 AM-11.30 AM	OOP KMS 227	OOP KMS 222	DM AIS 222	OOP KMS 222	DELD MDK 222	DM_R AIS 227
11.30 AM-12.15 PM	LUNCH BREAK					

Figure 3. Snap of Class Time table

5. CONCLUSION

By automating the timetable generation process using Google Sheets, organizations and institutions can save significant time and effort, leading to more efficient scheduling and resource allocation. This paper aims to demonstrate the feasibility and effectiveness of using Google Sheets as a versatile tool for automatic timetable generation.

6. REFERENCES

- Burke, E.K., De Causmaecker, P., Vanden Berghe, G., & Van Landeghem, H. (1999). A genetic algorithm approach to university timetabling. *Applied Artificial Intelligence*, 13(3), 263-280.
- Schaerf, M. (1999). A survey on timetabling problems: A classification, benchmarking, and analysis. *European Journal of Operational Research*, 119(3), 688-706.
- Prosser, P. (1993). A constraint programming approach to the nurse rostering problem. *Methods of Operations Research*, 61, 328-337.
- Abramson, D., & Ashtiani, S. (2007). A hybrid genetic algorithm for the nurse rostering problem. *Journal of Heuristics*, 13(5), 455-470.
- Di Gaspero, L., & Schaerf, A. (2002). An effective hybrid algorithm for university timetabling problems. *Annals of Operations Research*, 117(1-4), 283-305.
- Burke, E.K., & McCollum, B. (2007). A review of school timetabling problems and solution approaches. *European Journal of Operational Research*, 176(2), 649-669.
- Akshay Puttaswamy, H M Arshad Ali Khan, Chandan S.V, Parkavi. (2018), A Study on Automatic Timetable Generator, IJSIET.
- Y Ravi Raju, Mayank Mangal, (2017), Web-based Application For Automatic Timetable Generation, *International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)* ISSN: 2456-236x Vol. 02 Special Issue 03.
- Runa Ganguli, Siddhartha Roy (2017) A Study on Course Timetable Scheduling Using Graph Coloring Approach, *International Journal of Computational and Applied Mathematics*. ISSN 1819-4966 Volume 12, Number 2.