Automated Timetable Generation System Using Graph Colouring Technique

Shweta Gajbhiye[1] Ashwini Nagrale[2] Diksha Gondane[3] Diksheet Thul[4]
[1]Assistant Professor [2,3,4]Student
[1,2,3,4]Department of Computer Science & Engineering
[1,2,3,4]Priyadarshini J.L College of Engineering - Nagpur, Maharashtra 440009, India

Abstract— The traditional hand operated technique of time table is extremely time intense and typically lands up with frequent sets incompatible either at same part or with same academics having over one category at a time that is being resolved by Automated timetable Generator. This paper introduces a sensible timetabling approach using graph colouring technology capable of taking care of each hard and soft constraints needed specially for making ready plan in faculties with generous amount of academics and restricted resources like class-rooms or labs. It's conjointly capable to point out timetable as per ones need i.e. class wise, teacher wise, lab wise, room wise.

Key words: Graph Colouring Algorithm, Timetable, Hard Constraints, Soft Constraints, Scheduling

I. INTRODUCTION

In day to day life and also most probably people work related to organizations has faced some form of timetabling issues. The process of constructing time table manually for schools, colleges is very time overwhelming and requires lots of work as we have to take care of various constraints and preferences given by various teachers and resources that not been properly utilized. In order to eliminate all these drawbacks to produce a satisfactory result we develop an automated timetable generation system. The user enter for various inputs like total number of subjects to be taught, total number of teachers available, subject limits given by each teacher, subject preference given by each teacher, etc. and by taking all these inputs it will generate possible time tables making optimal use of all resources provided to it. The timetable must satisfy a number of requirements and desires of all people as much as possible. In a college, different courses are available, so there is no conflict of free timeslots available for every student within that time. Therefore a faculty member tries to find the timetable with the minimum chances of conflicts. A suitable timetable is then chosen from the optimal solutions generated. Timetabling is defined as a task to create a timetable without violating various constraints. Maximum and minimum work load of the teacher for a week will be stated for the effective generation of timetable. When picking a faculty as substitute it permits viewing timetable of that faculty to ensure that the faculty is free at that specific period. Substitute can approve or reject request. Principal can also view the request send by faculty and can also view substitute response.

It is a complete timetable management solution for Colleges which help to overcome the challenges in manually setting the timetable. It will be very easy for faculty to get timetable by using this software. We have also provided the feature to view the class time table, faculty wise time table and Lab Wise Time.

II. PROBLEM STATEMENT

The timetable creation problem is a combinatorial optimization problem that consists of four finite sets:

(*i*) a set of meetings, (*ii*) a set of available resources (e.g. rooms, labs, staff, and students), (*iii*) a set of available time–slots, and (*iv*) a set of constraints.[8]

The problematic is to assign resources and time slots to each given meeting, while maintaining constraints fulfilled to the highest possible amount. Automated Timetable Generator is a scheduling problem where each of the students' needs to attend the given data consists of a set of students and courses. A course is a set of actions that need to take place in the timetable. The above description of the Automated Timetable Generator defines a broad range of problems, whose complexity considerably depends on the specific constraints defined. Particular timetabling applications are usually focused on a more strictly defined subset of the problems, as the constraints and dimensions of the problem vary among institutions. We use the same approach, giving a detailed formal description of the problems for which our application is designed.

III. SCHEDULING & GRAPH COLOURING

In a classic semester, the courses are required to be planned at different times in order to avoid conflict. The graph colouring problem is the problematic way of describing a practical number (or the minimum number) of time periods needed to schedule all the courses subject to boundaries is a In graph theory, the graph colouring algorithmic program may be a special case of graph labelling; it's an assignment of labels historically referred to as "Colours" to parts of a graph, subject to certain constraints. In its simplest type, it's some way of colouring the vertices of a graph such no two adjacent vertices share an equivalent colour. This is often referred to as a vertex colouring. Graph colouring is one in every of the foremost vital ideas in graph theory and is employed in several real time applications in computer science.

The resolution of using colours invents from colouring the countries of a map, where each face is accurately coloured. This was global to colouring the faces of a graph embedded in the plane. By planar duality it became colouring the vertices, and in this form it simplifies to all graphs. In mathematical and computer representations, it is typical to use the first few positive or nonnegative integers as the "colours". In general, one can use any finite set as the "colour set". The nature of the graph colouring problem depends on the number of colours but not on what they are.

Graph colouring enjoys several sensible applications likewise as theoretical challenges. Beside the classical sorts of issues, totally different limitations may be assault the graph, or on the method a colour is assigned, or

perhaps on the colour itself. It's even reached quality with the overall public within the kind of the favoured range puzzle Sudoku. Graph colouring remains a really active field of analysis.

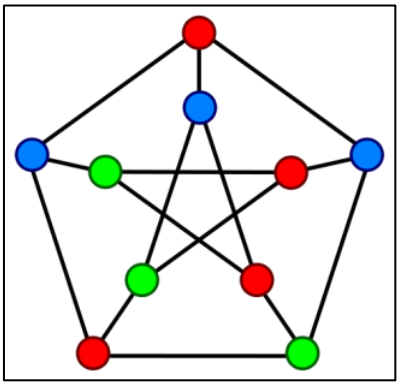

Fig. 1:

A. Definition of Chromatic Number

A graph G= (V, E) is k-colourable if there is exist a function c: V Æ {1, 2, … k} (the colouring function) so that if (a, b) OEE, then c(a) ! c (b) - that is, adjacent nodes must have "different colours". The least number k so that G is k-colourable is called the chromatic number of G, written c (G). The minor amount of colours needed to colour a graph is known as its chromatic number. The graphs, say graphs G with chromatic number is 1 (c(G) = 1) are the graphs which consisting entirely isolated nodes i.e. disconnected graph— if there will single edge also, we must have chromatic number as 2 (c(G) = 2). To find out a chromatic number of a graph is actually a NP-Complete problem. That means finding a chromatic number is belongs to NP-Complete problem category. NP-Complete problem is the part of computational theory. NP-Complete problem is a decision problem when it falls in NP and NP-Hard. We can denote the problems which are belong to NP Complete as NPC or NP-C. The chromatic number for a graph G is most commonly denoted as, but we can also denote it as. The chromatic number for a small graph can be computed by using Chromatic Number [G] by mathematical package Combinatory. We can use backtracking for minimum vertex colouring to compute the minimum colouring. The chromatic number of a graph should be equal to or may be greater than to its clique number. If the sub graph, say gi, has the chromatic number equal to the largest number of adjacent vertices in gi, then the given graph is Perfect Graph.

By explanation, it is given that the chromatic number of the line graph L (G) and the edge chromatic number of a graph G are equal. If a graph is having chromatic number as two, then it is known as Bi-colourable. And if chromatic number is three, then we call it as a three colourable. If a graph is having chromatic number as two, then it is known as Bi-colourable. And if chromatic number is three, then we call it as a three colourable. If we see in a general perspective, a graph which is having chromatic number as k, then it is said to be the k-chromatic graph, & a graph which is having chromatic number less than or equal to k is said to be k colourable.

IV. PROPOSED APPROACH

In order to treat timetabling problems we are proposing a system which would automatically generate timetable for the organization. Courses and lectures will be planned in contract with all possible constraints and given inputs and thus a timetable will be generated.

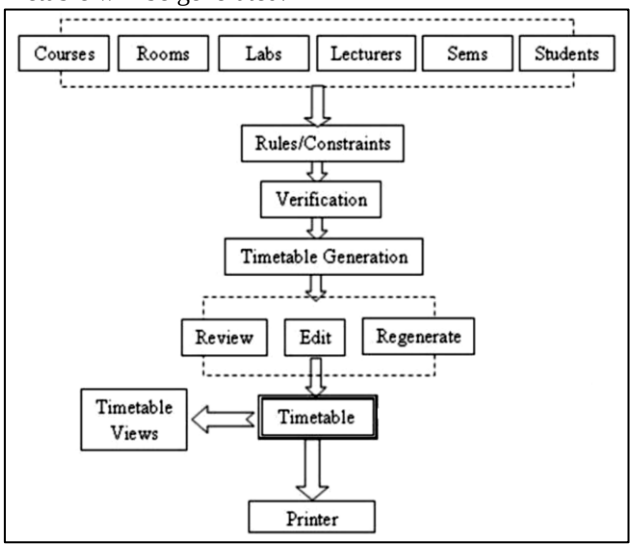

Fig. 2: Overall view of Timetable Generator [5]

Timetable construction was dull process in the previous approach of handover each subject to staff manually and creating the Timetable in such a way that no clashes occur. But this process acquired great use of time and also use of paper-work which is cost-ineffective. For this methodology we decide a solution of using our computing skills and technology to generate the Timetable. It is prepared using the Automated Timetable generator which contains the contribution of Graph colouring algorithm. The above Solution gives a block model of following processes: The user will enter each of the records as counts of subjects, class-rooms, labs, lectures, students. The admin will allocate each subject to their corresponding staff and assign those classrooms and the students whom they will teach. The Admin will use constraints as specified in the algorithm so that no conflicts occur. After assigning the Admin will do a verification check so that no inconsistencies are missed out. If the Admin encounters any mistake or clash that had been gone ignored earlier, he/she has the option to edit and then regenerate. After successful reviews the Timetable is ready for the staffs and students to view.

A. Constraints

The method of Timetable generation using the Automated Timetable generator comprises of the processing and Handling Data as the two stages in it. These two stages form a block where input will be set to these blocks and ideal output (Timetable) will be produced. The input includes the details of the number of subjects to be allotted, offered number of Lecturer to be given particular subject so that no clashes occur and also offered number of classrooms to be allotted to each subject. The Automated Timetable generator will take into consideration the inputs as subject-wise, staff-wise and class-wise. The inputs entered go through the processing stage first where there are operations like "Allocating subjects to time slots" and "Eliminate clashes"

where the calculations of fitting each data will be done using Automated Timetable generator. Then the Managing Data stage comes where the data of each subject, class and staff is taken to avoid any redundancies and generate the Timetable. There will be an allotted Admin who will take care of entering the inputs and the application will only be under the admins control to do any changes in the Timetable.

B. Hard Constraints of the Timetabling Problem

Sr. No	Constraints
1	No students can attend more than one lecture at a time
2	No lecturer can teach more than one subject at a time
3	No room can occupy more than one lecture at a time
4	Maximum number of time periods per day, that may not be exceeded
5	Events of the similar semester must not be assigned at the same time slot when both events are of kind "theory" or when one event is "theory" and one event is "lab practice". Same semester events can run simultaneously only if they are both of type "lab practice", as for each course 4...6 "lab practice" classes are planned within the week, each attended by a different group of students.
6	Specific lectures must be strictly assigned to specific teachers.

Table 1: List of Hard Constraints [9]

7

C. Soft Constraints of the Timetabling Problem

Sr. No	Constraints
1	The lectures are not allotted to time slots which come under the lecturer's prohibited time zones
2	The teaching hours on an instructor should be limited within the allowed maximum hours
3	A break for lunch must be allotted to the instructor
4	An appropriate lab with concern software should be allotted to specific subjects
5	Every teacher has a minimum and a maximum limit of weekly work-hours

Table 2: List of Soft Constraints [9]

D. Modules

We have proposed three modules

1) Insertion Module

In this module we deliver many user inputs to our system which acts as a fresh data for creating the absolute time table

2) Faculty Details

In this sub module we introduce various details of faculty such as faculty name, branch and their designation. We also provide a unique faculty id which supports in referencing during our software and it also acts login identifications.

3) Subject Details

In this sub module we introduce details of subjects that are in our course and their subject name. We attempt to store the theory subjects and lab subjects individually in the database so that it becomes easy for future use.

4) Planning

In this sub module we take user input that which faculty is taking which theory subjects and lab in a particular semester and we store in the database.

5) Allocation Module

In this module, user can pick any semester arbitrarily to start the method. Admin starts filling the time slots from the Monday by choosing the particular subjects; faculty that is mapped with that subject gets allocated to that time slot of the day. The various soft and hard constraints are tested every time the time slot is filled. We are testing all these constraints by writing query in stored procedure and using the database.

6) Display Module

In this module we can view the time table is generated of each class. We have also provided the feature to view the class time table, faculty wise time table and Lab Wise Time Table

7) Class Wise Time Table

In this we can view the class wise generated time table by selecting the particular semester which we want to see. The time table will have subject with faculty who handling that subject

8) Faculty Wise Time Table

In this we can view the entire subject name that is handled by a particular faculty**.**

9) Lab Wise Time Table

In this we can view the lab wise generated time table by selecting the particular semester.

V. CONCLUSION

The paper has presented a proposed approach of automating timetable generation by applying the graph colouring algorithm. The software will be fast, flexible, user friendly and having large storage capacity. It can deliver an effective preparation of courses and events where complex combinations of resources must be assigned efficiently to timetables. It will be helpful for the educational institutions and satisfy all their varying needs and will reduce the efforts of manual timetable generation as well as replaces the conventional software's or methods used for it. It can calculate the workload. It can test all the constraints applied and also check all the clashes and eliminate it.

REFERENCES

[1] Dr. Cauvery N K, "Timetable Scheduling using Graph Colouring." Vol. 1 Issue 2 - 2011.
[2] Baki Koyuncu , Mahmut Seçir, "Student Time Table By Using Graph Colouring Algorithm"
[3] Priyanka Gore "Timetable Generation Using Ant Colony Optimization Algorithm." Vol. 5, Issue 3, March 2017.
[4] Deeptimanta Ojha, Rajesh Kumar Sahoo, Satyabrata Das. "Automated Timetable Generation using Bee Colony Optimization." Vol. 10 – No.9, May 2016.
[5] Dipesh Mittal, Hiral Doshi, Mohammed Sunasra, Renuka Nagpure. "Automatic Timetable Generation using Genetic Algorithm." Vol. 4, Issue 2, February 2015.

[6] Spyros Kazarlis, Vassilious and Pavlina Fragkou. "Solving University Timetabling Problems Using Advanced Genetic Algorithm."
[7] Rajesh Kumar Sahoo, Deeptimanta Ojha, Durga Prasad Mohapatra, Manas Ranjan Patra,"Automatic Generation And Optimization Of Course Timetable Using A Hybrid Appraoch". Vol.95. No.1, 15 January 2017.
[8] Vatroslav Dino Matija, Goran Molnar, Marko _Cupi_c, Domagoj Jakobovi_c, and Bojana Dalbelo Ba_si_ "University Course Timetabling Using ACO: A Case Study on Laboratory Exercises".
[9] Shabina Sayed, Ansari Ahmed, Ansari Aamir, Ansari Zaeem. "Automated Timetable Generator." Vol. 1, Issue 11, April 2015.

