2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) | 978-1-6654-1451-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/Confluence51648.2021.9377151

Graph Coloring based Scheduling Algorithm to automatically generate College Course Timetable

P. Nandal
Dept.of Computer Science Engineering
Maharaja Surajmal Institute of Technology (GGSIPU)
New Delhi, India
priyankanandal@msit.in

Ankit Satyawali
Dept.of Computer Science Engineering
Maharaja Surajmal Institute of Technology (GGSIPU)
New Delhi, India
ankitsatywali4@gmail.com

Dhananjay Sachdeva
Dept.of Computer Science Engineering
Maharaja Surajmal Institute of Technology (GGSIPU)
New Delhi, India
dhananjay.sachdeva@gmail.com

Abhinav Singh Tomar
Dept.of Computer Science Engineering
Maharaja Surajmal Institute of Technology (GGSIPU)
New Delhi, India
abhinavtomar50@gmail.com

***Abstract*— Time table scheduling has been a topic of research in the past decades, yet, not a single optimal robust algorithm has been developed which can dynamically solve this problem considering all hard as well as some soft constraints. The manual way of doing it is very time-consuming, and even after generating a timetable, no one can guarantee that it is the best schedule. The automated timetable scheduling belongs to the NP-Complete class of problem, and hence it is not easy to solve. As academic institutes are growing in number and complexity, their resources and events are becoming harder to schedule, therefore to overcome all these problems, this paper proposes to make an algorithm which will consider all required constraints from both the students and teachers' point of view. The main aim of the proposed algorithm is to provide a conflict-free and practically feasible schedule with optimized use of every resource available.**

Keywords— Time Table scheduling, graph coloring, soft and hard constraints.

I. Introduction

Timetable scheduling is a big challenge faced by every academic institution. This activity is carried out by almost every academic institute once or twice a year. It is both time and man power-consuming job. Timetabling is the approach to use the limited resources most efficiently so that the desired goal can be achieved.

Gotlieb [1] was the first researcher who started the research to solve this traditional problem. As timetabling is an NP-hard problem, so its computational time grows exponentially as dataset size increases. Many solutions have been proposed till now, but research is still going on in this field. Initially, the research was focused upon school timetable, and therefore classical linear and integer programming approach was proposed but as the datasets and complexity of problems kept increasing these techniques were found inadequate due to this heuristic-based techniques were opted by researchers.

Many techniques are currently in use for solving timetable scheduling problems. Some of the more prominent techniques are graph coloring [2], backtracking [3], genetic algorithms [4], and local search procedures [5]. Also, there are various types of timetable scheduling problems such as:

• College Timetable

• School Timetable

• College/ School exam timetable

• Job scheduling

Graph coloring technique is used in real-time applications. It has applications in solving complex optimization problems particularly in conflict resolution and optimal partitioning of mutually exclusive events. Graph coloring technique is used in this paper to solve the timetabling problem. The fourth-year computer science department dataset of Maharaja Surajmal Institute of technology is used for testing the algorithm and also for measuring the efficiency of the algorithm. The algorithm presented in the paper considers various inputs like a list of classes, subjects, availability of teachers and classrooms, etc. and after processing all these constraints it generates a time table, making optimal utilization of all resources (teachers, rooms) in a way that will be best suited for all and satisfy all hard constraints.

The main aim of timetable scheduling is to generate a schedule that is not only conflict-free but also practically feasible. A timetable has to satisfy some constraints to be feasible. There are primarily two kinds of constraints that need to be satisfied by timetable scheduling: hard constraints and soft constraints. The constraints which are a must for an algorithm to satisfy are called hard constraints while the constraints which are not must but make the timetable more optimal and feasible are called soft constraints. General constraints satisfied by the approach presented in the paper are:

• A teacher can teach only be present in a single room at any time.

• Same room should not be allocated to the different classes in a specified time slot.

• Same subject having two lectures in continuity should have been allotted the same room.

Satisfying all the constraints in timetable scheduling is a very difficult task and so this study aims to satisfy all the required constraints by designing an algorithm based on graph coloring technique and produce a timetable that is conflict-free and practically implementable.

This paper is organized as follows. Section 2 describes the literature survey. The methodology is presented in section 3 and the results are described in section 4. The work done is then concluded in section 5.

II. LITERATURE SURVEY

The authors of [2] devise an algorithmic approach relating the time table scheduling problem with the vertex coloring problem in graph theory. By relating every course with a particular unique node, they try to assign unique slots to every subject. In the successive steps, they sort the subjects-slot combination in such a way, that no two subjects having the same slot are taught in the same room nor is the same teacher assigned to those two subjects. Using this, they are able generate a proper time table that takes care of all subjects yet the output is quite stringent as the lectures are allocated in a linear fashion. Thus, the main problem encountered by them is flexibility. In [6] the authors make a system in which the lecture schedule is made using the Welsh Powell graph colouring method. They use this method to differentiate between the lectures that can be conducted at the same time or at separate times by colouring each lecture node. Their approach helps generate a more time optimal solution but it too lacks flexibility, which is quite important while constructing timetables. [7] considers a course-conflict graph to which it applies graph coloring and then makes it conflict-free. The course-conflict graph is made by assigning the courses as vertices and the edges drawn between them represent conflicting courses i.e. courses with common students. This paper lays importance to the degree of constraint satisfaction, fair distribution of classes and optimal and unique solution. If multiple optimal solutions are present, then it chooses the one having the most discriminating conditions. [8] works on an existing feature of the graph coloring technique to generate clash-free timetables. It also considers several additional practical constraints and objectives. Multiple techniques based on Ant Colony Optimisation are then introduced and evaluated to identify the best solutions. Kristoforus et al [9] used the simulated annealing approach to solve the problem. Initially, a timetable is generated using a graph colouring approach and then all the violations present in the timetable are removed by iterating it again and again until an optimised version of the timetable is not obtained.

The algorithm used in [3] is based upon the backtracking paradigm. It applies the N-queen algorithm with a few changes. Firstly, one subject is considered and filled in the table so that not more than two classes are there each day. Then, other subjects are considered. If there is no exact time slot available, backtracking takes place. The authors of [10] propose an algorithm which is designed to consider the course preferences of the teachers in the institute. All the sections are assigned teachers according to their preference and if there are no preferences left, then randomly any section is assigned to the teachers. This algorithm doesn't guarantee an optimal solution but, moreover, it focuses on considering the soft constraints (teacher's preference of subjects). In [11] the authors consider the timetabling problem as a constraint satisfaction problem and propose a practical timetabling algorithm capable of taking care of both hard and soft constraints and finding variables' instantiation, based on the backtracking search. They consider a specific college as a use case that offers courses for both daytime and night time students. The constraints are formulated and arranged in the order of importance which helps in the variable instantiation. They also make use of a data structure to check the constraints. To improve efficiency, departments are ordered in some sequence, trying to find timetables sequentially according to the sequence, and then taking care of unscheduled classrooms. Anirudha Nanda and colleagues [12] present an adaptive approach to help solve the clashes of lectures and subjects. They develop a general solution from the teachers' point of view, and all constraints are set considering this fact. They initially create a random timetable which is optimized based on the heuristic of teacher availability. A clash data structure is maintained to ensure the same. To avoid any cycling and improve the accuracy randomization is used. A repetitive process ensures that all clashes are minimized, and an optimal solution is gained.

Salman Hooshmand et al [13] use Tabu search as a heuristic to develop the algorithm. They consider lessons as tuples and the required solution is set to be a pair of lessons and periods. To improve efficiency, two data structures are used in parallel and multiple objective functions are created to test the feasibility of constraints. The algorithm has three phases: Pre-scheduling, initial phase and optimization through tabu search. First, a graph-based algorithm is employed to create groups of lessons that have to be scheduled concurrently; then an initial solution is built using a sequential greedy heuristic. Finally, the solution is optimized using a tabu search algorithm based on frequency-based diversification. The authors of [5] propose an approach called particle swarm optimization (PSO). It is an evolutionary approach in which in the initial stage some specified amount of random schedules are generated and then the final schedule is evolved by comparing these randomly generated schedules based on various hard and soft constraints.

Ansari [4] uses a genetic algorithm to obtain a collisionless course schedule in an iterative process. He creates a chromosome containing a list of courses, rooms and time slot and evaluates the fitness value depending on the number of requirements violated. Their results show that due to a small initial population and less number of generations some conflicts remain in the solution. Thus, using genetic algorithms to get an optimal solution having fitness value of 1 the initial population and generations must be increased. Warke [14] makes use of a Genetic algorithm in combination with the Hungarian algorithm for task allocation. The genetic algorithm identifies and satisfies the hard constraints whereas the Hungarian model then optimizes the schedule with the least cost. Kolhe [15] uses genetic algorithms to create a semi-automatic timetable. They use hash tables to store the slot availability of each object along with its constraints. Their interactive two-phased approach creates an initial solution first, followed by a mutation on a selected solution to get the best result. The authors of [16] create an application in which an initial solution is devised using the n-queens algorithm and then genetic algorithmic approach is used to optimize the solution. They create an initial database and use an iterative approach by individually creating time tables for each year and then integrating them. They also give higher priority to practical and tutorial classes and schedule theory classes at the end. Their application can create a time table for multiple courses and semesters. The authors of [17] present a novel crossover technique to generate optimum schedules using the genetic approach. This paper customizes only one child in each crossover process.

Every gene of the offspring is taken from only the first and second parent based on the soft constraint penalties. The lower penalty is selected to form the new child. Some soft constraints have ignored that need to check the global schedule is ignored, and the crossover only considers the first and second parent and not the complete schedule from one parent.

III. METHODOLOGY

In this study graph coloring (vertex coloring) technique is used to solve the time table scheduling problem. In vertex coloring, the nodes are colored in such a way that no two adjacent nodes share the same color. The timetable scheduling problem is mapped with the vertex coloring problem in such a way that an optimal schedule can be generated from that. The schedule can be further optimized using the Welsh Powell algorithmic approach as done in the below-mentioned algorithm. Welsh Powell algorithm is a modified version of the graph coloring algorithm wherein nodes are first sorted in decreasing order and then processed to produce more optimal results.

A. Algorithm

1. Find the degree of each vertex.
2. List the vertices in order of descending degrees.
3. Assign different priorities to different types of subjects i.e. practical (maximum), tutorials and lectures (minimum).
4. Assign required slots to the subject in decreasing order of priority (from practical to lectures).
   4.1 Assign time slot satisfying all hard constraints.
   4.2 Assign rooms satisfying room conflict constraint.
5. Repeat step 4 for all the remaining subjects in the decreasing order of priorities.

In this algorithm, the subjects are related to nodes to form an undirected graph such that the edges between the nodes (subjects) represent the different types of constrains taken into consideration. This undirected graph is shown in the above figure – Fig. 1 and acts as an input for the algorithm. The algorithm starts by sorting the subjects (nodes) according to the decreasing priorities (primary condition) and in-degrees (secondary condition). Different priorities are set for different types of subjects such as practical labs have the maximum priority and lectures have the minimum. This sorted order provides a way to process the data in a more efficient manner with which a schedule can be generated in optimal time & space. Then come the time slot and room allocation part. Firstly, all the available slots are computed for the subject in consideration. This is done by eliminating those slots which are already assigned to the adjacent subjects. Then some random slot from the list is considered depending upon the flexibility of that slot. After time slot allocation, rooms are allocated too in the same fashion. After the algorithm processes the graph, a feasible schedule is displayed in a tabular manner, as shown in the tables below – Table 1, 2, 3, 4. The proposed algorithm is a greedy algorithm which in worst case takes O (n^2) time complexity and O (n) space complexity but doesn't always provide us with the most optimal solution but in the average case, we can expect a feasible solution.

B. Input

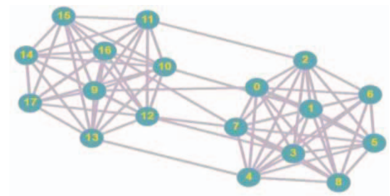

Fig. 1- Graph representing the subjects as nodes and edges as the constraints between them

Algorithm is taking following values as input:

- The List of subjects (S0, S1, …, Sn) that are taught to the students.
- List of lecturers (L0, L1, …, Ln) who are assigned to teach subjects.
- List of subjects (S0, S1, …, Sn) that are taught under the same course.
- List of subjects (S0, S1, …, Sn) that are taught by the same lecturer.
- List of laboratories (LA0, LA1, …, LAn) that shares common rooms.
- List of classes per week of every subject.
- List of rooms (R1, R2, …, Rn) that will be assigned to classes.
- List of time slots (T1, T2, …, Tn).

C. Constraints

The algorithm proposed in this paper is satisfying the following constraints:

- All subjects have to be assigned a time slot and rooms.
- No room is occupied by more than one class at a time.
- No lecturer is assigned more than one class at the same time.
- No student should have different classes at the same time.

D. Output

Table 1 – Subject time slot allocation for Class-1

Day/Timeslot	T1	T2	T3	T4	T5	T6
Monday	2	2	3	3	0	0
Tuesday	1	1	-1	-1	6	4
Wednesday	5	-1	-1	4	7	6
Thursday	-1	5	4	-1	7	-1
Friday	5	8	-1	7	6	-1

Table 2 – Subject time slot allocation for Class-2

Day/Timeslot	T1	T2	T3	T4	T5	T6
Monday	12	12	11	11	10	10
Tuesday	9	9	13	-1	13	17
Wednesday	-1	-1	-1	-1	-1	-1
Thursday	15	14	-1	15	-1	15
Friday	16	14	16	13	16	14

Table 3 –Room allocation for Class-1

Day/Timeslot	T1	T2	T3	T4	T5	T6
Monday	R3	R3	R4	R4	R1	R1
Tuesday	R2	R2	-1	-1	R5	R5
Wednesday	R5	-1	-1	R5	R5	R5
Thursday	-1	R5	R5	-1	R5	-1
Friday	R5	R5	-1	R5	R5	-1

Table 4 – Room allocation for Class-2

Day/Timeslot	T1	T2	T3	T4	T5	T6
Monday	R4	R4	R3	R3	R2	R2
Tuesday	R1	R1	R5	-1	R6	R6
Wednesday	-1	-1	-1	-1	-1	-1
Thursday	R5	R6	-1	R5	-1	R5
Friday	R6	R6	R5	R6	R6	R5

IV. RESULT AND DISCUSSION

The algorithm is tested on final year CSE Dept. dataset of Maharaja Surajmal Institute of Technology which consists of two classes (CSE-1 & CSE-2) with a total of 18 subjects (8 labs, 10 lectures) and 14 lecturers. The dataset has been constructed into an undirected graph given as input to the program. The output so obtained satisfies all the hard constraints whilst also considering some soft constraints. The output is complete and there are no clashes of time slots or room availability as shown in figures (1-4). Practical classes and tutorials have been given a higher priority and the schedule is generated accordingly.

The program is successful in generating a fully fledged timetable in less time, yet there are some limitations to it. The schedule generated does not guarantee optimality as there are some soft constraints which could not be effectively satisfied. All in all, the proposed approach helps generate a complete schedule which helps eliminate the manual process of making timetables and provides all stakeholders with a much better alternative.

V. CONCLUSION AND FUTURE SCOPE

A time table scheduling algorithm is designed by using the vertex coloring algorithm to generate the course schedule in optimal time and space complexity. This algorithm handles all the specified hard constraints i.e. one room will be allocated to one class at any time, no student will be assigned two different classes at the same time, no lecturer will be assigned two classes at a time, along with satisfying soft constraints such as no student will have same subject lecture more than once a day. The proposed algorithm is tested on the final year CSE dept. data set and is producing feasible results in optimal time and space.

This algorithm can be made more accurate and optimal with a better slot allocation function which will help in satisfying the soft constraints such as considering lecturers' preferences and providing free lectures after some specific consecutive classes, thus making this scheduling algorithm more flexible.

RERFERENCES

[1] Gotlieb, C.C "The construction of class-teacher timetables." In IFIP Congress, vol. 62, pp. 73-77. 1963.

[2] Avinash, Rahul Jain, Rajat Kumar, "University Time Table Scheduling Using GraphColoring Technique", https://www.researchgate.net/ , 2018.

[3] Abhinaya, V., K. Sahithi, and K. Akaanksha. "Online Application of Automatic Time-Table Generator." International Research Journal of Engineering and Technology (IRJET) 6, no. 02 (2019).

[4] Ansari, R., and N. Saubari. "Application of genetic algorithm concept on course scheduling." In IOP Conference Series: Materials Science and Engineering, vol. 821, no. 1, p. 012043. IOP Publishing, 2020.

[5] Abayomi-Alli, Olusola, Adebayo Abayomi-Alli, Sanjay Misra, Robertas Damasevicius, and Rytis Maskeliunas. "Automatic examination timetable scheduling using particle swarm optimization and local search algorithm." In Data, Engineering and Applications, pp. 119-130. Springer, Singapore, 2019.

[6] Abdullah, Dahlan, Marsali Yaton, Heri Sujatmiko, Sepyan Purnama Kristanto, Hendra Nazmi, Irma Lisa Sridanti, Andang Suhendi et al. Lecture Scheduling System Using Welch Powell Graph Coloring Algorithm in Informatics Engineering Departement of Universitas Malikussaleh." In Journal of Physics: Conference Series, vol. 1363, no. 1, p. 012074. IOP Publishing, 2019.

[7] Poddar, Narayan, and Basudeb Mondal. "An Instruction on Course Timetable Scheduling Applying Graph Coloring Approach." International Journal of Recent Scientific Research 9, no. 2 (2018): 23939-23945.

[8] Khan, Koffka, and Ashok Sahai. "Tabu Ant Colony Optimisation for School Timetable Scheduling Problem."

[9] Bendi, R. Kristoforus J., and Hadi Junaidi. "SIMULATED ANNEALING APPROACH FOR UNIVERSITY TIMETABLE PROBLEM." Jurnal Ilmiah MATRIK 21, no. 3 (2019): 204-213.

[10] Karmaker, Debjyoti, Mir Riyanul Islam, H. Rahman, A. Bhowmik, and M. N. Imteaj. "A heuristic approach to course scheduling problem." American International University-Bangladesh (2015).

[11] Hwang, Kyoung-Soon, Keon Myung Lee, and Joongnam Jeon. "A practical timetabling algorithm for college lecture-timetable scheduling." In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 817-825. Springer, Berlin, Heidelberg, 2004.

[12] Nanda, Anirudha, Manisha P. Pai, and Abhijeet Gole. "An algorithm to automatically generate schedule for school lectures using a heuristic approach." International journal of machine learning and computing 2, no. 4 (2012): 492.

[13] Hooshmand, Salman, Mehdi Behshameh, and Omid Hamidi. "A tabu search algorithm with efficient diversification strategy for high school timetabling problem." arXiv preprint arXiv:1309.3285 (2013).

[14] Warke, Yamini, Durga Munje, Abhijit Swami, Shweta Raskar, and Gauri Tapkir. "Automatic Timetable Generation using Genetic and Hungarian Model."

[15] Kolhe, Apurva D., Chaitali D. Patil, Mitali A. Aherrao, and Riyanka S. Rai. "Automatic Scheduling System using AI." IJETT 6, no. 2 (2019).

[16] Ambhore, Shraddha, Pooja Walke, Rohit Ghundgrudkar, Akshay Alone, and Anushree Khedkar. "Automatic Timetable Generator."

[17] Saptarini, Ni Gusti Ayu Harry, Putu Indah Ciptayani, and Ida Bagus Irawan Purnama. "A Custom-based Crossover Technique in Genetic Algorithm for Course Scheduling Problem." TEM Journal 9, no. 1 (2020): 386-392.