

Automatic Scheduling System using AI

Apurva D. Kolhe, Chaitali D. Patil, Mitali A. Aherrao, Riyanka S. Rai

Department of Computer Engineering,

NDMVPS KBTCOE, Nashik-422013 ,

Savitribai Phule Pune University, Maharashtra.

apurvakolhe.ak@gmail.com, 29chaitalipatil@gmail.com, aherraomitali@gmail.com, riyankarai31@gmail.com

Abstract—The traditional method of manually creating a timetable is very tedious and time consuming, due to this many times different problems occur like various classes may clash, extra load on certain teachers and this all may ultimately affect the curriculum of students. To overcome all these problems we propose an automated timetable scheduling system. This system will be capable of handling both hard and soft constraints which are specially required for preparing timetables in colleges where there is a large number of students and limited classrooms, labs, and faculties as well as limited time to complete the curriculum considering the exam period. We propose a system which will take faculty details, allocated subjects and number of classrooms and labs available as well as hours allocated by the university as an input. Once the final timetable is generated, separate timetables for faculty, labs and different classes will be available as an output. It will also keep a track of previous year's generated timetables.

Index Terms—Agile, Genetic Algorithm, Hard constraints, Random Adaptive Search, Soft constraints.

I. INTRODUCTION

Today almost every faculty and institute needs a timetable to execute any work in a timely and orderly fashion. Especially in educational institutes where there are different courses, a large number of students and few teachers. The traditional method of creating a timetable includes a staff member who is responsible for creating the timetable; he/she would manually create the timetable and try satisfying the required structure. Now usually the person would refer the previous years timetables and with that would create the new one. One problem with this is that the same old format of timetable would be repeated again and again with some minute changes. This would be unfair to that faculty who would have the same time slot, again and again, every semester or year. We would primarily look into engineering colleges, where there are different branches and many times for some classes the faculties of different branches are also required. In this case, it becomes very difficult to efficiently create a timetable which would satisfy all the needs. This process would take a considerable time of the faculty as well as efforts which rather could have been used at another important task. It also requires a lot of paperwork, and in a time like today where there is such advancement in reducing human efforts in almost every field then why not here?

The timetable scheduling is considered as a Constraint Satisfaction Problem (CSP) in Artificial Intelligence because to generate a single timetable we need to satisfy many different constraints or conditions, only by which we would be able to

generate an efficient timetable for the required job. Even during scheduling simplest and smallest constraint can cause a huge problem because of which an optimal timetable is even harder to create and find. Hence this problem is also an NP-Complete problem.

Automatic Scheduling System is a Java-based software used to generate the university timetable automatically. The system would help to manage all the periods automatically. It would make sure that no faculty is given extra load in a day or treated unfairly. The system will need different inputs like the details of staff, rooms, labs, subject allocated to the staff as well as time allotted to each subject from the university. The constraints to generate the timetable will already be set according to the specific institute and can be changed later onwards if institute requests for it. These constraints can be soft as well as hard. Hard constraints are those which the system has to satisfy and cannot generate a timetable without them. Soft constraints can be avoided if they are violating any hard ones. The system would try to satisfy all the hard constraints and maximum of the soft ones. While dealing with the leave request, the system will check which faculty can substitute the faculty on leave and try to minimize their load. The system will display the list of prospects and the admin has to decide which faculty will be best suited for that subject and the system will generate the new timetable. Selecting a faculty as a substitute it allows viewing timetable of that faculty to ensure that the faculty is free at that particular period.

II. HISTORY & BACKGROUND

Timetable generation problem is a highly constrained one, but above all the problem differs greatly for different colleges, institutions, and others. It is hard to generate the timetable for all of these diverse applications based on a single problem. There exist a lot of different timetabling problems like university timetable, exam timetable, school timetable, sports timetable, and worker timetable. Also, there exist different approaches to problem-solving which use meta-heuristics like a genetic algorithm, Backtracking, Constraint Logic Programming. In recent years two major approaches have proved beneficial namely Local Search Procedures and Constraint programming.

A. Heuristic Procedures

Different heuristic methods are used for solving computationally hard optimization problems within a reasonable time frame which is good enough for the problem. Local Search algorithms like a genetic algorithm, the travelling salesman problem are also heuristic in nature. These algorithms move from solution to solution in the space of candidate solution by applying local changes until an optimal solution is found. Heuristic methods do not guarantee that the solution found will be optimal [1][5][6].

B. Constraint-Based Approaches

One major advantage of constraint programming is that clear definitions of the constraints serve beneficial for solving the given problem. This enables the program easy to adjust. The main disadvantages of this approach are:

- 1) The difficulty with expressing soft constraints
- 2) The potential problems that occur while enhancing the initial basic solution.

The capability to convey composite constraints in a simple, declarative way is critical for establishing the specification of the colleges and university timetable problem into the program and is critical for their successful result [6].

C. Constraints

There are two types of constraints namely Hard and Soft Constraints. Hard constraints are those constraints which while solving any problem cannot be violated in any case. We can obtain the desired solution when all of these hard constraints are satisfied. On the other hand, the maximum of the soft constraints satisfaction will be the goal [5]. Different constraints for the timetabling problem are as follows: Basic mandatory constraints:

- 1) At a particular slot a teacher, classroom or lab cannot be allocated more than once.
- 2) During the break time, no lecture should be scheduled.
- 3) Teacher's working hours cannot exceed the institute timings.
- 4) A load of every teacher must be balanced weekly.

Some Hard Constraints are as follows:

- 1) A Teacher must teach a particular class every morning for a certain time.
- 2) A specific days working hours for a teacher is fixed.
- 3) Class Ys hours are from 5 pm to 7 pm Monday to Friday.
- 4) Room R is not available on Saturdays.

Some examples of secondary constraints are:

- 1) If possible, total working hours for a teacher between 8 am to 5 pm should not exceed 4 hours.
- 2) If possible, a teacher should not be required to come in the morning at 8 am more than 3 days a week.
- 3) If possible, there should be at least a 30-minute gap between any two lectures for every teacher.
- 4) If possible, a Teacher or student need not come to college for a single lecture.

III. DESIGN

There are different methods which can be followed to get a feasible solution for the timetabling problem. One approach is to create a semi-automatic timetable. To create effective timetable requires hybrid approach and therefore the semi-automatic two-phased user interactive approach is discussed by Aditya Bhatt and Lakshmi Kurup [2] that first creates an initial population and then the mutation is done on a selected solution in the second phase by combining best approaches. There is extensive use of hash tables in this approach. Hash tables are used to store slot availability of each entity and the constraints associated with it. Here entities are nothing but the rooms, labs, faculties, etc. They not only simplify the development but also help in running the algorithm faster as the time required to search for availability is reduced. The Hash Tables can be used to store the unavailability or availability of different entities. For these, availability hash tables will look like this

For an entity Room R:



Slots from 6 to 8 are removed and slots from 11 to 16 removed

For Lectures L:



Slots after 6 and before 9 are removed because the lecture L cannot be allotted in those slots. Another famous method to solve the timetabling problem is to use the genetic algorithm completely or to use in combination with other algorithms.

Genetic Algorithms are inspired by Darwin's evolutionary theory. GA comes under the class of Evolutionary algorithms that use the principle of natural selection to derive a set of solutions towards the optimal solution. It is a search heuristic which generates solutions to optimization problems using techniques inspired by natural evolution like mutation, inheritance, crossover, and selection [1].

Initially, the genetic algorithm starts with a set of solutions which are called as the population. Solutions from one generation of the population are taken and then used to form a new generation of the population. This is done in hope that the new population will be better than the old population (stronger). The new solutions are selected based on their fitness to form new solutions (offspring); suitable solutions will have a higher chance to produce more suitable solutions and so on. This is repeated until all the constraints are satisfied.

Algorithm: Genetic Algorithm (GA)

BEGIN

INITIALISE population with random candidate solution.

EVALUATE each candidate;
 REPEAT UNTIL (termination condition) is satisfied DO

- 1) SELECT parents;
- 2) RECOMBINE pairs of parents;
- 3) MUTATE the resulting offspring;
- 4) SELECT individuals for the next generation;

END.

The reasons to use a genetic algorithm for finding the fittest solution are:

- 1) There are multiple local optima.
- 2) The objective function is not smooth (so derivative methods cannot be applied)
- 3) The number of parameters is very large
- 4) The objective function is noisy or stochastic
- 5) A large number of parameters can be a problem for derivative-based methods when you don't have the definition of the gradient. In this type of situation, you can find an acceptable solution through GA which can be improved with a derivative based method.

GA is able to find solutions in very less time. Here the solutions are solutions which are good according to the defined heuristic. Due to the random mutation, we can obtain a wide range of solutions.

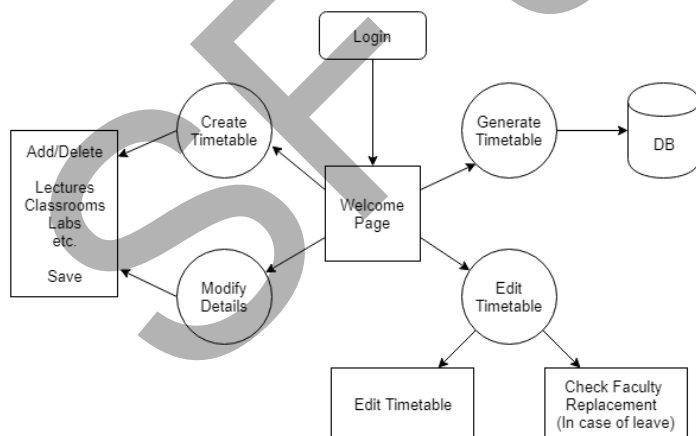


Fig. 1. . Block Diagram of the proposed timetable scheduling system.

IV. RESULT AND ANALYSIS

The resulting system would have a model as shown in figure 1. Advantages of the system

- 1) Eliminate Paper-based process: Manual creation of timetable involves a huge amount of paperwork as well as manually scheduling the timetable which would involve many errors. Due to this, delays may occur and waste resources as well as the time of the institute.
- 2) Intuitive & User-friendly: It is simple and easy to use. Very less technical knowledge is required to operate it. It is very easy to implement in any institutions of any size.
- 3) Customization & Flexibility: Timetable software can be fully customized to meet the unique class scheduling needs and suggestions of the institution
- 4) Optimal Resource Allocation: Assign teachers and classrooms for periods and optimize the allocation of resources in the best manner possible.

Limitations

- 1) It may not be fair in terms of slot generation for all students.
- 2) Incomplete data provided by admin would lead to
- 3) Invalid Constraint specification may lead to faulty algorithm generation.

V. CONCLUSION

The proposed system concludes that for automated generation of timetable many different approaches are available like meta-heuristic methods as well as constraint satisfaction methods. Depending on the user's interaction with the system and the system's requirements any algorithm can be followed or a hybrid one can also be created. Thus to get an optimal solution different algorithms can be used at different phases.

REFERENCES

- [1] Deeksha C S, A Kavya Reddy, Nagambika A, Akash Castelino, K Panimozhi, "Automatic Timetable Generation System," Journal of Emerging Technologies and Innovative Research, 2015.
- [2] Aditya Bhatt, Lakshmi Kurup, "Interactive Selection of Timetables Generated Using Evolutionary Multi-Objective Optimization, Institute of Electrical and Electronics Engineers, 2017.
- [3] Anuja Chowdhary, Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, Dinesh Gawande, "Timetable Generation System," IJCSMC, Vol. 3, Issue. 2, February 2014.
- [4] Anirudha Nanda, 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, Vol. 2, No. 4, August 2012.
- [5] Prof Er. Shabina Sayed, Ansari Ahmed, Ansari Aamir, Ansari Zaeem, "Automated Timetable Generator," International Journal for Innovative Research in Science & Technology Volume 1 Issue 1 April 2015.
- [6] Akshay puttaswamy, H M Arshad Ali Khan, Chandan S.V, Parkavi. A, "A Study On Automatic Timetable Generator," International Journal Of Science And Innovative Engineering & Technology