International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

Design and Implementation of Algorithm for Time Table Generation

Kevin Menezes[1], Vikas Jha[2], Krishna Jaiswal[3], Niket Amoda[4]
[1,2,3]*Student, Department of Electronics and Telecommunication Engineering, Thakur College of Engineering and Technology, Mumbai, India*
[4]*Assistant Professor, Department of Electronics and Telecommunication Engineering, Thakur College of Engineering and Technology, Mumbai, India*

***Abstract*: Time table generation is tedious job for educationalist with respect to time and man power. Providing an automatic time table generator will help to generate time table automatically. Proposed system of our project will help to generate it automatically also helps to save time. It avoids the complexity of setting and managing Timetable manually. In our project we are going to use algorithms like genetic, heuristic, resource scheduling to reduce these difficulties of generating timetable. These algorithms incorporate a numeral of strategy, aimed to improve the operative of the search operation. The system will take various inputs like number of subjects, teachers, and workload of a teacher, semester, and priority of subject. By relying on these inputs, it will generate possible time tables for working days of the week for teaching faculty. This will integrate by making optimal use of all resources in a way that will best suit the constraints.**

***Keywords*: generation, educationalist, automatically, managing, resource scheduling, time table.**

1. Introduction

Although majority college organization work has been mechanized, the lecture timetable preparation is still commonly done by hand due to its inherent difficulties. The physical lecture-timetable preparation demands significant time and efforts. The manual lecture-timetable scheduling is a limitation fulfillment problem in which we find a result that satisfies the given set of constraints. There have been numerals of approaches made in the earlier period to the difficulty of constructing timetables for colleges and schools. Timetabling problems may be solve by diverse methods inherited from operation study such as graph coloring, local search measures such as tabu search, simulated annealing, genetic algorithms or from backtracking based constraint fulfillment handling. In our project, timetable problem is formulated as a constraint fulfillment problem and we proposed a realistic timetable algorithm which is capable of taking care of both hard and soft constraints. It is a complete timetable solution for Colleges which help to overcome the challenges in manually constructing the timetable.

2. Literature survey

A. Timetabling

It is large and highly constrained, but above all the problem differs greatly for diverse colleges and learning institutions. It is hard to write a universal agenda, fitting for all possible timetable problems. Even though manual creation of timetable is sustained, it is still universal, because of the lack of suitable computer programs.

Timetable problems

There exist a lot of diverse timetable problems such as:

- University Timetable
- Exam Timetable
- School Timetable
- Sports Timetable
- Worker Timetable

Moreover, there exist a lot of problem solving methods, which typically use the concept of customary optimization algorithms such as genetic algorithms, Backtracking, Constraint Logic Programming.

In recent years two major approaches appear to have been victorious.

- Local Search Procedures
- Constraint Programming (CP)

B. The Local Search Procedures

The local search measures such as Simulated Annealing, Tabu Search and Genetic Algorithms. These methods convey constraints as various cost functions, which are minimized by a Heuristic Search of enhanced solution in a neighborhood of some opening realistic result.

1) Simulated annealing (SA)

Simulated annealing is a probabilistic method used for similar to the global optimum of a given function. Purposely, it is a metaheuristic to fairly accurate global optimization in a huge search space. It is frequently used when the search space is distinct. Simulated annealing is a technique for finding a good result to an optimization dilemma. If there is a condition where we want to maximize or reduce something, our problem can likely be tackle with simulated annealing.

2) Tabu Search

Tabu Search is a Global Optimization algorithm and a Metaheuristic or Meta-strategy for calculating a surrounded heuristic method. Tabu Search is a parent for a huge relations of derivative approach that establish memory structure in Metaheuristic, such as Tabu Search and Parallel Tabu Search.

3) Genetic algorithm (GA)

Genetic Algorithms (GA) was imaginary by John Holland and has described this thought in his book "Adaptation in natural and artificial systems" in the year 1975. Genetic algorithm is a metaheuristic motivated by the procedure of natural selection that belong to the bigger class of evolutionary algorithms (EA). Genetic Algorithms are motivated by Darwin's evolutionary theory. GA comes below the class of Evolutionary algorithms that use the principle of natural collection to develop a set of solution towards the best result. It is a search heuristic which generates solutions to optimization problems using technique motivated by natural evolution like mutation, inheritance, crossover and selection.

C. GA Operators

1) Chromosome representation

Genetic material is a set of parameter which describe a planned result to the problem that the genetic algorithm is demanding to answer. The genetic material (chromosome) is often represented as a easy string. The fitness of a chromosome depends upon how well that chromosome solves the problem at hand.

2) Initial population

The first step in the performance of a GA is the production of an initial population. Each member of this population encodes a potential solution to a problem. Each unit is evaluated and assigned a fitness value according to the fitness function. It has been acknowledged that if the initial population to the GA is good, then the algorithm has an enhanced option of finding a good result and if the initial supply of construction blocks is not large enough or good enough, then it would be hard for the algorithm to find a good result.

3) Selection

This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be chosen to reproduce. During each successive production, a portion of the accessible population is selected to selection a new generation. Individual solutions are chosen through a fitness based process, where fitter result are usually more likely to be chosen.

4) Crossover

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one creation to the next. It is parallel to reproduction and organic crossover, upon which genetic algorithms are based. Crossover takes more than one parent solutions and producing a child solution. There are techniques for collection of the chromosomes. Crossover arbitrarily exchanges the subsequences before and after that locus between two chromosomes to create two children. The crossover operator roughly does as it is natural recombining between two single chromosome organisms.

5) Mutation

Mutation is used to sustain genetic diversity from one creation of a population of genetic algorithm chromosomes to the next. It is parallel to natural mutation. Alteration (mutation) alters one or more gene values in a chromosome from its initial situation. In mutation, the result may alter totally from the previous result. Hence GA can come to enhanced result by using mutation. Mutation can take place at each bit position in a string with some possibility, usually very small.

6) Fitness Function

The fitness function is described over the genetic representation and procedures the quality of the represented result. The fitness function is forever problem reliant In particular, in the fields of genetic programming and genetic algorithms, each design result.

After each round of testing, the thought is to remove the 'n' worst design solution. Therefore, desires to be awarded a shape of merit, to signify how close it came to meeting the general necessity, and this is generated by applying the fitness function to test, results obtained from that solution.

7) Heuristics

It finds out solution amongst all possible ones, but they do not prove that the most excellent will be found, they may be thought as about and not correct algorithms. These algorithms, usually find a solution close to the most excellent and they find it quick and merely. These algorithms can be correct, that is they convert and find the best solution, but the algorithm is stationary called heuristic until this best solution is proven to be the best.

Disadvantages of local search procedures are:

- The complexity of taking into relation of hard constraints.
- The need to decide their parameters through testing even though they are excellent for optimizing the initial realistic solution.

D. The Constraint Programming (CP)

Major advantage of constraint programming is declaratively a clear-cut statement of the constraints serves as part of the program. This makes the program easy to adjust, which is critical in timetable constraints are handled through a system of constant propagation, which minimizes domains of variables, coupled with backtracking search. In modern CP languages, both features do not need to be planned explicitly.

The main disadvantages of this approach are:

1. The difficulty with expressing soft constraints.
2. The potential problems with enhancing the initial feasible solution.

The capability to convey composite constraints in a simple, declarative way is critical for establishing of the colleges and university timetable problem into the program and is critical for their successful tailored delivery approach is able to introduce soft constraints during a search, leading quickly to a "Good"

timetable, integration of local search into CP gives the capability to optimize efficiently the timetable.

E. Constraints

There are a variety of constraints to be satisfies at the time to instantiate variables about time slots and classrooms. The constraints can be categorized into Hard and Soft constraints.

1) Hard Constraints

A timetable which breaks a hard constraint is not a feasible solution. Hard constraints comprise Conflicts"

- HC1. A classroom is not assigned to more than one teacher at the same time.
- HC2. A teacher cannot teach more than one class at the same time.
- HC3. Courses for the similar year-session students of a department cannot take place at the same time.
- HC4. The classroom for a course should have enough capacity to take students registered in the course.
- HC5. The classroom should be well set of equipment with required services for the classes.

2) Soft Constraints

Soft constraints are less significant constraints, and it is typically not possible to avoid breaking at least some of them. Either timetable functional, which calculates the level to which a timetable has violated its soft constraints. Some soft constraints are more vital than others, and this is often specified priority value.

- SC1. The teachers are not assigned to time sl in the teacher's prohibited time zones.
- SC2. Teachers daily teach hours should be within the acceptable utmost hours.
- SC3. As far as possible, classes are scheduled in the lecturer's preferred time zones.
- SC4. A lunch break must be scheduled.
- SC5. The practical courses are scheduled in morning session, and the theory courses are sch session.
- SC6. The lecture hours for a course should be scheduled consecutively.
- SC7. As distant as possible, classes should be corresponding department's exclusive- use classrooms
- SC8. The classrooms should be allocated in a manner to reduce the distances between adjacent class's classrooms. It is desirable for timetables to satisfy all hard and soft to meet all these constraints. Despoiled in any case, but some soft constraints can be sacrificed to find reasonable is made harder by the fact that so many people are affected by its outcome.

3. System design

Design of a scheme is basically a blue print or a plan for the system. It determines the part structure of the components. Design contains the following things:

1. First year timetable it contains timetable of first year based on that we will create higher semester timetables.
2. Faculty details in department tells that the details of respective faculty in department.
3. A workload detail tells that the higher and lower workload that faculty has based on their designation. That is for professor has less work than the assistant professor.
4. Subject details as subject name, subject code.
5. Faculty and subject allotment table consist for which subject respective faculty is allotted based on timeslots.
6. Theory and lab courses related details contain the details of each subject that is handled by respective faculty.

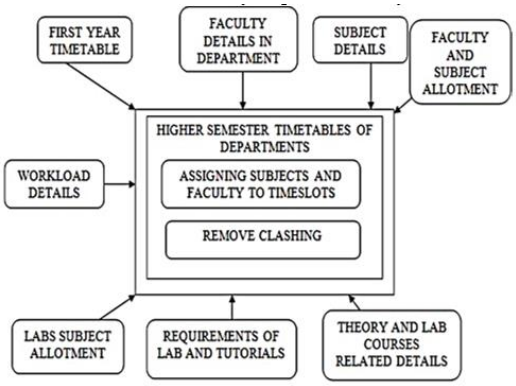

Fig. 1. System design of timetable

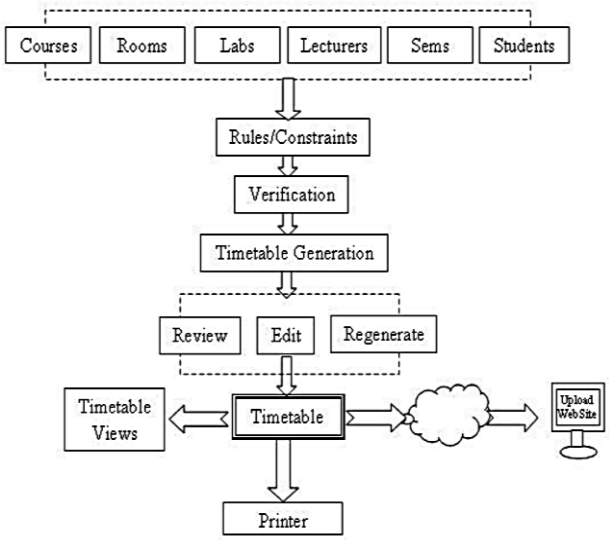

Fig. 2. General view of TTGS

A. TTGS Features

- Simple...Easy-to-understand & simple-to-operate with intuitive user interface, guides and samples
- Fast...Generates timetable in minutes, saving your time, efforts and money.
- Rules configuration and verification
- Different views of timetable depending upon who is viewing timetable with Timetable Generation System, timetable generation process involves just a few simple steps.

4. Scope of project

Timetable Generation System automatically generates timetable for each class, laboratories and teachers, in keeping with the availability of teachers and lab assistance, workload of teacher and lab assistance, Reporting time of the staff, availability and capacity of available physical resources (such as classrooms, laboratories and computer room) and rules abided at different classes, semesters, teachers and subjects according to the level .Best of all, this Automatic Timetable Generation System tremendously and efficiently improves resource utilization and optimization.

5. Methodology

A design methodology is a methodical approach to creating a plan by applying a set of methods and guiding principle. We have followed these methodologies.

- Total requirement of the system including the framing of timetable strategy should be concerned A database should be formed. As for every rules taken for the reason of maintaining the records. Record all possible scenarios and then upcoming with flow-charts to handle the scenario. The scheme should be carefully tested by running all the test cases written for the system.
- Firstly, we should do timetables for first year classes so by entering details of first year timetables of all section. Read the details like faculty, subjects, section into the database.
- Retrieve the first year timetable timeslots and assign the timeslots to the respective faculty.
- Retrieve the subjects allotted for first year faculty and start filtering their subjects in the timetable.
- The faculty who handles the theory classes will be allotted for tutorial and related lab.
- All faculties should get two hours of classes in the day session depending on the subject allotted.
- The workload allotted for professor, associate professor, and assistant professor have to be followed.

6. Functional components of the project

Following is a list of functionalities of the system.

- Slots are assigned for lab and counselling hours.
- Faculties are assigned classes; each has an interval of at least 1 Period.
- Faculty is assigned maximum of 2 lectures in a day or 1 lecture and 1 practical session in a day. In worst case only one faculty may have 3 lectures in day.
- Subjects can be of any of the following categories:
    1. Compulsory subjects.
    2. Department Electives.
- Each Faculty workload - 10 hours / week (Theory), - 8 hours /week (Practical).
- Faculties assigned with Elective Subjects workload 8 hours/week (Theory) and 9 Hours/week (Practical).
- Maximum & minimum of subjects should be specified.
- Preference is given to the other department staff while assigning slots.

7. Conclusion

It is complicated task that to handle many Faculty's and allocating subjects for them at a time physically. So our proposed system will help to overcome this disadvantage. Thus we can produce timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools are widely applicable and free to use also.

References

[1] Bagul, M. R., Pushkar R. Patil, S. C., & Nagare, S. N, "A Novel Approach for Automatic Timetable Generation," in International Journal of Computer Applications, vol. 6, October 2015.
[2] Chowdhary, A., Priyanka Kakde, S. D., & Rupal Rushiya, D. G, "Timetable Generation System," International Journal of Computer Science and Mobile Computing, vol. 6, February 2015.
[3] Deshkar, M., Mayur kale, M. B., & Ghom, A., Time Table at a Click," in International research journal of engineering and technology, vol. 4, March 2016.
[4] Lahoti, Y., & Aaditya Punekar, H. P., "Automated Timetable Generator," in International Journal of Science and Research, vol. 4, 2015.
[5] M, S., & Pranav Kiran Vaze, P. M., "Automatic Time Table Generator. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, no. 5, May 2017.
[6] Mittal, D., and Hiral Doshi, M. S., "Automatic Timetable Generation using Genetic Algorithm," in International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, February 2015.
[7] P. Okunieff, "Dynamic Timetable Generator," Final Report for Transit IDEA Project 39, Transportion research board of the national academics, March 2006.
[8] Patil, M. K., & Rakhe Shruti Subodh, P. A, "Web Application for Automatic Time Table Generation," in International Journal of Current Engineering and Technology, vol. 3. June 2014.
[9] Rathod, P. P., Kamlesh K. Lodhiya, M. P., 'Automatic Timetable Generator," in International Journal of Research in Science & Engineering, vol. 8, 2016.
[10] K. Panimozhi, A. Kavya Reddy, A. Nagambika, Akash Castelino, and C. S. Deeksha, "Automatic Timetable Generation System," Journal of Emerging Technologies and Innovative Research, vol. 2, no. 4, pp. 1006-1012, April 2015.