Timetable Handling Mechanism Using Python

Manpreet Kaur, Jasdev Bhatti, Mohit Kumar Kakkar, Deepika Goyal

Abstract: In this paper timetable strategy is designed and discussed in detail for handling any university course scheme based on resources available. In addition to regular courses some new additional parameters like Engineering exploration, Team teaching technique, mentoring of students, etc. are discussed that run parallel with unique mechanism. The allocation of timetable on basis of three different modules has been discussed satisfying all major and minor challenges. The language named python is applied for optimizing and displaying timetable into three different views class wise timetable (for students), faculty timetable, and classroom timetable.

Index Terms: Team teaching scheme, Course scheme, Constrains, Faculty load, Python.

1. INTRODUCTION

Before the beginning of a session the first and foremost thing that is of the biggest concern to any department and the concerned faculty is his/her time table. Time table is the arrangement of the classrooms and the classes in an appropriate manner to get the definite schedule matching the requirement of the university curriculum. To every university, time table is a unique concept to make coordination in between the faculty, students, parents and the other academician. Although nowadays every university has its own manual method or software usage technique for making time table but making a perfect time table is still a cumbersome task. Any software can arrange the cards (lectures) according to the input only but giving it a perfect input/shape requires days of planning by department. However, once the input is given and the cards are arranged then also the arrangement of the cards is done manually. Hence this paper is studied to develop the practical approach for building lecture course timetabling system, which can be customized to fit to any colleges timetabling problem. The college lecture-timetabling problem asks us to find some time slots and classrooms which satisfy the constraints imposed on offered problem. In our proposed problem, the problem is not just fitting a normal course scheme of the first-year students to the time table but also keeping in mind the special courses and the arrangements provided by the university to the first-year students, keeping many minor and major constraints in mind. As per to the regular courses followed by educational institutions in timetable there is an involvement of few new courses like Engineering Exploration which is also divided into six different verticals. The important factor to be notice is that all students who had enrolled in six different verticals have to appear to each one vertical at the same time in groups but in different venues. Along with this the new concept of team teaching in few subjects has also been inculcate in our proposed timetable.

- *Manpreet Kaur, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. Email: manpreet,kaur@chitkara.edu.in*
- *Corresponding Author*: Jasdev Bhatti, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. Email: jasdev.bhatti@chitkara.edu.in*
- *Mohit Kumar Kakkkar, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. Email: mohit.kakkar@chitkara,edu.in*
- *Deepika Goyal, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. Email: deepika.goyal@chitkara.edu.in*

In 90's time tables were handled manually, and every department had their individual time table in which there was no constraint about the faculty involvement in different departments like Bachelor and Master Courses Mathematics, Physics, Chemistry etc. Yang, Y. et al (2006) had studied the general solution model for the course timetabling problem. In 2007 Oprea, M. had contributed by studying a multi-agent system for university course timetable scheduling. In 2012 Al-Betar, M., et al. presented an algorithm named "Harmony search algorithm" which is a metaheuristic population-based algorithm. This algorithm was derived from natural phenomena of musician's behavior, that they co-operatively play their musical instruments for achieving fantastic harmony. They applied same algorithm to university course timetabling against standard benchmarks and explained its capability of providing a viable solution compared to those in previous works. Further, Al-Betar, M., et al. discussed a technique named "Memetic computing technique" that is designed for UCTP, called the hybrid harmony search algorithm (HHSA). In HHSA, the harmony search algorithm (HSA), which is a metaheuristic population-based method, has been hybridized by two factors: first the hill climbing, to improve local exploitation and the second one is a global-best concept of particle swarm optimization to improve convergence. The results were compared against 27 other methods using the 11 datasets of Socha et al. comprising five small, five medium, and one large datasets. The proposed method achieved the optimal solution for the small dataset with comparable results for the medium datasets. Furthermore, in the most complex large datasets, the proposed method achieved the best results. Narang, B. et al (2013) had contributed by making use of active rules and genetic algorithm to generate the automatic time-table In 2014, Babaei, H., et al, had contributed to university course timetabling problem by studying the approach based on multi agent systems (cooperative search) in addition to briefly study approaches based on operational researches, meta-heuristic methods and intelligent novel methods. Genetic algorithm to generate the automatic time-table –an over view is analyzed by Ansari, A., Bojewar, S. in 2014. In the same year, Chowdhary, A., et al, introduced a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automated timetabling system. In his study each teacher and student can view their timetable once they are finalized for a given semester, but they can't edit them. Ottoum, I. in 2015 given the result of the analysis and synthesis processes that take place when making a timetable for a University Information System (UIS), especially for SIS. Along with this he compares between two methods of designing a time table and shows the advantages and disadvantages of these

methods and more precisely the implementation of each using programming languages. Mittal, D. et al (2015) had given an automatic timetable generation using genetic algorithm. Kralev, V. and Kraleva, R. (2016), had studied a local search algorithm based on chromatic classes for university course timetabling problem. In last year 2017 Babaagba, K., Arekete, S. carried out an extensive survey of various mobile agent systems that have been utilized in solving university course time tabling problem. He discussed about the number of methods that have been adopted in solving the university course time tabling problem such as the use of linear numberings, evolutionary algorithms, graph coloring among others. They studied one recent technique employed involves mobile agents that are simply software agents which move from system to system within the network transporting both their state and code. These are not bound to the system where their execution begins. Also, Saritha, M., et al, has proposed to make an automated system that will take various inputs like details of students, subjects and class rooms and teachers available, depending upon these inputs it will generate a possible time table, making optimal utilization of all resources in a way that will best suit any of constraints or college rules. List of subjects may include electives as well as core subjects. Also, Khiarak, J., et al, investigated the appropriateness of using the genetic algorithm (GA) and the imperialist competitive algorithm (ICA). The proposed technique consists of two steps: (i) using the proposed manipulated GA for solving the timetabling problem of each department, and (ii) eliminating the interference of common events among multiple departments and satisfying the hard and soft constraints by using ICA. Finally, a report on the efficiency of the methodology used in this study was obtained from the University of Tabriz in Iran and University of Udine in Italy. In this paper, the results are revealed in two ways: (i) reduction in the problems due to shrinking of the database and solving of the problems in parallel and (ii) solving the different parts of the problem by using various criterion results, increasing the common events satisfaction in that sub-problem. Eventually, the proposed model provided successful satisfaction of the hard constraints in <700 iterations with GA and elimination of interference in 40 iterations with ICA in most of the cases. In all previous research the timetable is managed manually or by some software. But no one has given value to optimizing the timetable mechanism. By this paper, timetable strategy is designed and optimized in detail for handling any university course scheme based on resources available. In addition to regular courses some new additional parameters like Engineering exploration, Team teaching technique, mentoring of students, etc. are discussed that run parallel with unique mechanism. Python language is used for optimizing and displaying timetable into three different views class wise timetable (for students), faculty timetable, and classroom timetable satisfying all major and minor challenges.

.

2. SALIENT FEATURES OF THE SYSTEM

- Automatic timetable software (ASC) generate timetable automatically.
- An efficient time table is proposed according to availability of resources and requirement.
- Management of faculty load according to their need and availability.
- Satisfying all needs and constraints/challenges of departments and management.

3. ADDITIONAL PARAMETERS

As per to the regular courses like Physics, Chemistry, Mathematics, Electronics, electrical followed by educational institutions in timetable there is an involvement of few new courses like Engineering Exploration (divided into six different verticals) and mentoring for counseling session and interaction of students.

3.1 Team Teaching

As we know that every faculty is having specialization in some subtopics of any subjects. So, in order to give the students different or the best techniques of teaching, grouping of faculties with topic expertise has been prepared before timetable. For example, the two groups having six faculty with three in each has been divided for two different chapters that to be covered up before first sessional. Each group has been assigned with one topic and classes, which are going to be swapped after getting over with one topic. Same methodology has to be followed up in between first and second sessional. In this way, a teacher is rotated in different classes according to their expertise in the topic being taught.

3.2 Engineering Exploration

As per today industrial demands for engineering students is the extra project and innovation knowledge of student except his/her academic skills. In such university has taken the initiative by introducing the new course named “Engineering Exploration” to the first-year engineering students for all branches like computer, electronics, electrical and mechanical. This course was subdivided into six different specializations in which students were divided according to the student’s online registration. The course was integrated with electronics, language and mechanical tools to buildup overall engineering development of students. In this faculty of different specialization or department are involved in subgroups for each six categories. The important thing to be care of is that all fifteen classes and faculty have to be assigned for same slots, different venues and six different courses.

3.3 Labs

Engineering first year students study Engineering chemistry Lab, Engineering physics Lab and PSTC (Problem Solving techniques using C) which are two hours each and a class is divided into two groups for these labs. Now when half the class is attending any of these labs then simultaneously the other half group of the class can only have a lab or a tutorial parallel to these labs, which is of the biggest concern for the faculty making the time table.

3.4 Mentoring

A college mentorship program is a service intended to give guidance to upcoming freshmen and transfer students. In order to successfully transition from high school into college. Each class is divided into two groups and each group has a mentor to it which can guide them through different phases of their college life. As there is a special lecture for the same so we must give a special room for the same for each group of a class.

3.5 Genetic Algorithms (GA).

We know that Time Table Scheduling problem is representative of the class of, NP-hard, combinatorial optimization problems with multi-constraints and for this we

are using genetic algorithms (GA). GA has been found to be capable of getting optimized solutions for a number of problems like Time Table Scheduling for which no other algorithm exists. It is inspired by nature as altering a population of candidate solutions until a maximum or minimum solution is found. Scheduling a Timetabling faces two types of constraints, namely soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which if we violate them; the timetable is no longer valid. The search space of a timetabling issue is excessively tremendous, numerous arrangements exist in the search space and some of the arrangements are not possible. Possible arrangements here mean those which don't damage hard constraints and as well attempt to fulfill soft constraints. We have to pick the most suitable one from possible arrangements. Most suitable ones here mean those which don't disregard soft constraints to a more noteworthy degree. In this paper hard- constraints have been dealt with carefully and it has been guaranteed that soft constraints are too pursued however much as could be expected.

4. PROPOSED SYSTEM

The final system should able to generate time tables in completely automated way which will save a lot of time and effort of an institute administration. To make a timetable system generic so that everyone can work equally well for different School, Colleges and Universities, the system had been divided into four modules:

- Insertion module
- Allocation module
- Algorithm
- Display module

4.1 Insertion module

There are certain requisites that must be fulfilled for the data entry in the same way. There are the following requirements:

a) The name of the faculty from each department.
b) Classrooms, labs details and its capacity.
c) Subjects (theory, tutorials and labs) details with its total number of lectures

But there are certain pre requisites for the data entry of the software which are:

i) Load of the faculty must be decided in advance.
ii) The course scheme must be well made

Now, with this are certain constraints and challenges that have to be taken care of while making the load of the faculty and making the load. The Challenges faced during compilation of load for BE First Students can be categorized into two parts:

- Minor Challenges (Soft-Constraints)
- Major Challenges (Hard-constraints)

4.2 Minor Challenges

- A teacher cannot attend two classes at the same time.
- A course cannot be taught in two different classes at the same time
- A teacher may request for a special classroom for a given course
- The courses should be scheduled in a way that empty time slots of both teacher and students are minimized.
- The lunch break should either be at 11:45 a.m. or at 12:40 p.m.
- Up to some extent equal load should be given to all faculty members.
- Limit time (hours per week) is given to every Batch

4.3 Major Challenges

- Faculty involved in teaching is from different departments so both the time tables should be checked at the same time.
- During Team teaching, the teacher is to be swapped after every Sessional Exam.
- As the teachers are going in different departments so they cannot have lectures in different departments continuously.
- As students are studying PSTC Lab which they are taught by a single teacher in two different groups so while one group is studying PSTC then at that time the second group students can be occupied in some other lab or tutorial.
- There should not be any clashes between two faculty members and two class rooms/venues. At the same time period there should not be a clash between two lectures of a class section
- The biggest challenge is the infrastructure where the availability of classrooms as in our proposed problem, the number of lecture halls is 9 and tutorial rooms are 2.
- As the students are involved in Engineering Exploration in which 2 sections are combined according to their strength so those two sections must have their Engineering Exploration class at the same time with six different categories and venues.
- The Engineering Exploration class should be of 2 hours from 9 to 11 or from 12:40 to 2:30 pm.
- Four teachers are assigned in Engineering Exploration class at the same time.
- The rooms of Engineering Exploration class should be fixed.

4.4 Allocation Schedule

This module is the most important one among all three, as under this module the user compile and start entering the inputs that he/she received from all departments. The procedure of entering the details follows the pattern:

- Subjects were assigned according to course scheme with
  teaching hours load (Labs, Theory, and Mentoring).
- Numbers of section/ classes were assigned with time slots of 55 minutes per lecture.
- Faculties were assigned of all departments with teaching load.
- All entries were divided according to day and slot wise.

Now with the above all inputs, the software arranges the timetable accordingly. After the completion of data by software the timetable incharge must shuffle the slots of lecture keeping all minor and major challenges.

4.5 The algorithm

Initialization

{this step will create a population of N individuals, satisfying for every individual a set of hard and soft constraints:

Soft-Constraints:

• Up to some extent equal load should be given to all faculty members

• limit time (hours per week) is given to every Batch
Hard-constraints (rigid):
• There should not be any clashes between two faculty members and two class rooms/venues

• at the same time period there shold not be a clsh between two lectures of a class section

}
while (End Test not Verified)
do {the end test is on the number of iterations performed}
begin
Reproduction implementation;
Crossover implementation;
for l:=1 to N do begin apply mutation of order k;
Mutation implementation;
if (LOCAL_ON) then apply local search {LOCAL_ON may be 0 or 1}
if (no. of infeasibilities > max no. of infeasibilities) then apply filter
end;

end.

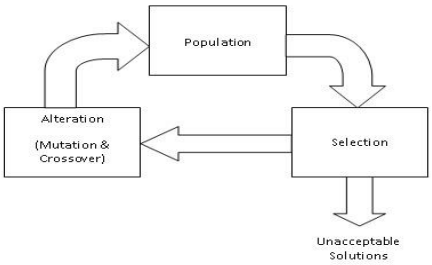

Figure 2: *Algorithm Evolutionary Cycle*

Parameters
Size of Population = 150
probability (Mutation) = 0.035
Probability (Crossover) = 0.9
Maximum Generations = 500

4.6 Display Module
Under this module the incharge can view how the timetable is finalized in display mode for each class and faculty. The software has also benefitted us by providing the time table in three different formats:

- Class wise timetable
- Faculty timetable
- Classroom timetable

4.7 Class wise timetable
The class wise time table is mainly generated for display to students. This view is subdivided into the following characteristics:

a. Class for which time table assigned
b. Abbreviation used for the faculty assigned for class with subject.
c. Labs assigned for class.
d. Venue assigned for lecture.
e. Parallel engineering exploration assigned to class with name of faculty and venues.
f. Mentoring lecture with name of mentor and venue.
g. Tutorials assigned in parallel to labs with assigned groups.

Lecture Hall

	TG005	TG006	TG007	TG008	TG011	TG101	TG102	TG103	TG104	TG105
9:00			CH103 - CSE1			GE101 - CSE 2			AM101 - CSE 1	
10:00			CH103 - CSE1			GE101 - CSE 2			AM101 - CSE 1	
11:00			CH103 - CSE1			GE101 - CSE 2				
12:00						GE101 - CSE 2				
13:00										
14:00										
15:00										
16:00										

Class- CSE 1A

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
9:00	CH103 - TG007	CH101 - TG006	EG103 - TG011		CS101 - TG101
10:00	CH103 - TG007	CH101 - TG006	EG103 - TG011		CS101 - TG101
11:00	CH103 - TG007		EG103 - TG011		CS101 - TG101
12:00			EG103 - TG011		
13:00					
14:00					
15:00					
16:00					

Figure 2: *Class wise Timetable*

4.8 Faculty timetable
This view is generated for displaying the teaching load of faculty. This view is subdivided into the following characteristics:

a. Class for which faculty is assigned
b. Name of faculty for which teaching load is prepared.
c. Abbreviation of subject assigned to teacher.
d. Venue assigned for lecture.
e. Engineering exploration class with two different sections assigned to faculty with venues.
f. Mentoring lecture with assigned mentoring group and venue.

Figure 3: *Faculty Timetable*

Professor Name

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
9:00	GE101 - TG101	CH101 - TG006			PH103 - TG006
10:00	GE101 - TG101	CH101 - TG006			PH103 - TG006
11:00	GE101 - TG101				PH103 - TG006
12:00	GE101 - TG101				
13:00					
14:00					
15:00					
16:00					

4.9 Classroom timetable
This view is generated for displaying the classroom wise timetable required for department regarding availability of rooms for any other activity other than academics or for any other extra lectures to students and any internal evaluations. This view is subdivided into the following characteristics:

a. Class for which room is assigned with timings and lecture number.
b. Abbreviation used for the faculty assigned for class with subject
c. Abbreviation of subject assigned to class with lecture number.
d. Venue assigned for lecture.
e. Engineering exploration class for two different groups with assigned all faculties, venues.
f. Mentoring lecture with assigned mentoring group, faculty as mentor and venue.

The major advantage of this the software been discussed in this paper is that for any mistake or changes due to team

teaching or by any other reason, the timetable incharge must change it in any single one of the three views. But the software changes it automatically in other two views that depicts it time saving nature.

Parameters	Iteration-I	Iteration-II	Iteration-III	Iteration-IV
Size of Population	150	150	150	150
probability (Mutation)	0.035	0.035	0.035	0.035
Maximum Generations	500	500	500	500
Probability (Crossover)	1.00	0.95	0.75	0.50
Population fitness	0.217	0.0210	0.0188	0.0154

Figure 4: *Classroom Timetable*

5. CONCLUSION

This paper has developed the approach of an automated time table system that helps in creating the timetable in three different categories: faculty, class wise (for students) and classroom wise timetable. The structural view of timetable generated were impressive by providing information to all applicant concerned to it. This paper had managed the data or resources with all necessary details and then frames a perfect lecture wise timetable as compared to manual generation of timetable involving the satisfaction of all challenges i.e. no overlapping, time saving, easy to handle, avoiding continuation of lectures for faculty, fixing any parameter easily to any challenges.

REFERENCES

[1] Yang, Y., Paranjape, R., Benedicenti, L., (2006) An agent based general solution model for the course timetabling problem, Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, May 08-12, Hakodate, Japan, pp.1430-1432.

[2] Picard,G., Bernon, C., Gleizes, MP., (2006), ETTO: Emergent Timetabling by Cooperative Self-organization, International Workshop on Engineering Self-Organising Applications, pp.31-45.

[3] Oprea, M., (2007), MAS_UP-UCT: A Multi-Agent System for University Course Timetable Scheduling, International Journal of Computers Communications & Control, 2(1), pp. 94-102.

[4] Al-Betar, M., Kader, A., Gani, T., (2012), A harmony search algorithm for university course timetabling, Annals of Operations Research, 194(1), pp.3-31.

[5] Al-Betar, M., Khader, A., Zaman, M., (2012), University course timetabling using a hybrid harmony search metaheuristic algorithm, IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews), 42(5), pp.664-681.

[6] Narang, B., Gupta, A., Bansal, R., (2013), Use of active rules and genetic algorithm to generate the automatic time-table, International Journal of Advances in Engineering Sciences,3 (3), pp.40-44.

[7] Babaei, H., Hadidi, A., (2014), A review of distributed multi-agent systems approach to solve university course timetabling problem, Advances in Computer Science: An International Journal, 3(5), pp.19-28.

[8] Ansari, A., Bojewar, S., (2014), Genetic algorithm to generate the automatic time-table –an over view, International Journal on Recent and Innovation Trends in Computing and Communication,2(11), pp.3480-3483.

[9] Chowdhary, A., Kakde, P., Dhoke, S., Ingle, S., Rushiya, R., Gawande, D., (2014), Time table generation system, IJCSMC, 3(2), pp.410 – 414.

[10] Ottoum, I., (2015), The Analysis of the time table structure within a student information system (sis), International Journal of Computer Science & Information Technology (IJCSIT) 7(1), pp.77-83.

[11] Mittal, D., Doshi, H., Sunasra, M., Nagpure ,R., (2015), Automatic timetable generation using genetic algorithm, International Journal of Advanced Research in Computer and Communication Engineering,4(2), pp.245-248.

[12] Kralev, V., Kraleva, R., (2016), A local search algorithm based on chromatic classes for university course timetabling problem, International Journal of Advanced Computer Research, 7(28), pp.1-7.

[13] Babaagba, K., Arekete, S., (2017), A review of agent-based university course time tabling systems, IJEDR, 5(2), pp.566-568

[14] Saritha,M., Vaze, P.,Pradeep, Mahesh N R,(2017), Automatic time table generator, International Journal of Advanced Research in Computer Science and Software Engineering,7(5),pp-204-211.

[15] Khiarak, J., Harghalani, Y., Derakhshi, M., (2017), Combined multi-agent method to control inter-department common events collision for university courses timetabling, Journal of Intelligent Systems, pp. 1-17