

# University Timetabling based on Hard Constraints using Genetic Algorithm

Sanjay R. Sutar

Associate Professor, Department of IT  
Dr.B.A.Tech.University,Lonere,MS,India

Rajan S. Bichkar

Research Guide, CSE  
S.R.T.M.University.Nanded, MS, India

## ABSTRACT

The proposed system presents a novel approach of solving University timetabling which is NP-hard problem using Genetic Algorithm. Genetic Algorithm is frequently deployed Meta heuristics algorithm which can be effectively used to difficult combinatorial optimization problems. Although, there has been an extensive research towards this field but majority of the research results are much in its nascent stage. The previous researchers have used various methods like Tabu search, Simulated Annealing, network flow, graph coloring, etc. Genetic Algorithms are effective in solving many such optimization problems. The current work presented uses Genetic Algorithm to design an effective model for scheduling with challenging constraints considerations. The objective of the research is to create a model using Genetic Algorithm to the extent it can be used to generate the acceptable schedule using probabilistic operators like mutation and crossover. The design of the fitness function has considered the hard constraints. The simulation shows the better result in minimum time.

**Keywords** Class scheduling problem, cross over, genetic algorithm, mutation.

## 1. INTRODUCTION

University curriculum timetabling is considered as one of the NP hard problems [1] which are computationally complex to resolve. The quantity of essential computation in order to explore an optimal result actually amplifies exponentially with the structure of the problem. The main purpose of class scheduling or timetabling process includes satisfying certain set of constraints with respect to variables like lecture rooms, number of lecturers, time slots, number of branches, number of subjects, number of semesters etc. The problem of timetabling is being under observant eyes of researchers since two decades but unfortunately no 100% optimized results has been published in any literature. The reason for this is the dynamic scenario of constraints and datasets used for the experiments. Various institutions will have various protocols for maintaining their timetabling for their students, which will not match exactly to some other universities or any other institutions. Therefore, it is almost near to impossible to design a uniform framework or model which will work for all the datasets. It was found in many situations that course curriculum timetabling will consists of exploring the schedules which does not violate all the restraints. Such category of issues is described as search problems [2]. Many

other scenarios, it is also found that problems under consideration are described as optimization problems [3]. Such situation calls for satisfying all the hard constraints and reduces or increases a considered objective function which entrench the soft constraints. There are also other techniques where the optimization method will act as a media to implement optimization on search problems. Timetabling problem cannot be completely automated according to many researchers. There are various reasons for it, e.g. one schedule can be better than other due to input of datasets to be experimented. Also, as the search space is normally large, manual management may bias the search in much better optimization. Majority of the prior research work has concentrated to solve timetabling using Artificial Neural Network [4], Fuzzy Logic [5], Hill climbing [6], Operation Research [7], Simulated Annealing [8], Tabu search [9] and Genetic Algorithm [10]. The issue has become so high that from past 10 years, institutions like PATAT [11], University of Twente [12] have been conducting Timetabling competition where a large number of researchers worldwide participate to provide solution. The proposed system highlights University timetabling for a real-time dataset, which has considered much challenging scenario of hard constraints. The results are optimized using Genetic Algorithm to show better timetabling for one week timeslots.

The paper is organized as follows, Section-II introduces the past research work done. In section III we represent the main problem formulation related to course curriculum, brief introduction of proposed Genetic Algorithm. Section IV discusses the results and analysis, followed by conclusions in Section V.

## 2. RELATED WORK

The problem of automated generation of efficient timetabling system has been under keen study for research for many years. This section will attempt to visualize the effective research work done with proper classification of techniques as below:

A huge research has been conducted in the field of timetabling using Simulated Annealing technique. Edmund Burke et al. have mechanized Simulated Annealing technique for analyzing the behaviour of diversified examination

timetabling when it is optimized from preliminary results provided by a graph-coloring heuristic with backtracking. Ruibin Bai et al. [13] researched on hyper-heuristic technique which works on search space of heuristics and also uses stochastic heuristic selection strategy with a short-term memory. Tunchan [14] has worked towards faculty-timetabling problem considering lecturer seniority scale. Nandhini [15] has conducted an intensive survey for implementing Simulated Annealing method for solving timetabling problem. Koyuncu [16] has worked on automated student schedules generation using graph coloring. Naseer et al. [17] designed Roulette wheel Graph coloring heuristic for solving examination timetabling problems. Cauvery [18] has also attempted to redesign a timetabling issue using graph coloring and Ant algorithm. Fen [19] has researched on constraint based reasoning technique in order to solve the complexity of University Course Timetabling Problem (UCTP) by searching the optimized preferential value, dependent on student capacity for each lecture. The constraint based technique is always in search for better timeslots and lecture hall for the specific lecture. Unfortunately, constraint based approaches are found to be computationally expensive because of the exponential growth of feasible assignments with variables considered. Ozcan [20] has investigated an incremental strategy for solving the problem by approximating to a better solution in a rational time which increases for a large set. Nguyen [21] has highlighted a new model using very difficult multi-objective scheduling problem with a memetic algorithm. Unfortunately, it has been seen that it takes long enough to show the results. Gaspero [22] has conducted an investigation on a family of Tabu Search based techniques whose neighborhoods concerned to which contributed to the violations of hard or soft constraints. They designed a Tabu Search methodology for exam timetabling where ordered priorities were given for the constraints. The constraints were considered in two ways: (1) one constraint at a time from the highest priority, where ties were broken by considering the lower priority constraints; (2) all the constraints at a time, starting from the highest priority. Cordeau et al. [23] proposed a Tabu Search meta-heuristic to solve the timetabling problem designed by Ben Paechter [24] for the International Timetabling Competition organized by the Metaheuristic Network and sponsored by PATAT (Practice and Theory of Automated Timetabling). Gunawan [25] has presented a framework for solving timetabling in University of Indonesia consisting of allocation of teacher to the course and course sections by amalgamating Tabu search with Simulated Annealing technique. Datta and Carlos [3] had research on the same issue considering datasets from IIT Kanpur using evolutionary algorithm for multi-objective type of problems. Nurmi [26] has presented a framework for highly constrained school timetabling problem, which was modeled from the requirements of various Finnish school levels. But still the authors are pretty not sure of the efficiency when the constraints are shuffled dynamically.

Research on bi-objective evolutionary techniques with a motive of reducing length of timetable and spacing out the

conflicting exams was done by Cote, Wong, and Sabouri [27]. A novel design of mathematical model on standard Genetic Algorithm for resolving examination timetabling problem in training centers with a motive of incrementing the intervals between the exams was done by Sheibani [28]. Erben [29] has designed a grouping Genetic Algorithm in which a precise encoding and fitness functions were analyzed.

### 3. PROPOSED SYSTEM

The proposed system considered a real time dataset of timetabling of Dr. Babasaheb Ambedkar Technological University, Maharashtra, India. The proposed system aims at designing an optimized curriculum timetabling for one week considering Monday to Friday as working days using Genetic Algorithm. The considerations of the proposed research work are as tabulated below:

**Table 1 Parameters under consideration**

|                                       |         |
|---------------------------------------|---------|
| Number of Branches                    | 3       |
| Number of Semesters in each branch    | 8       |
| Number of Students in each branch     | 60      |
| Number of Subjects in each semester   | 5       |
| Number of Lecturers in each branch    | 20      |
| Number of class rooms for each branch | 5       |
| Class room capacity                   | 60      |
| Time duration for 1 session           | 1 hour  |
| Working time                          | 9am-6pm |
| Lunch break                           | 1pm-2pm |

The soft constraints may be as follows:

1. In laboratory room, it is feasible to conduct theory session.
2. A specific session may be required to be provisioned in a specific time period.
3. Lecturers may choose to have all their sessions in a number of days and to conduct number of sessions free days.
4. Each lecturer has their minimum and maximum limit of weekly working hours.

The hard constraints of the proposed work are as follows:

1. It is not possible to conduct laboratory session in a classroom with no laboratory facility.
2. One lecturer can take only two specific subjects related to specific branch and semester.
3. Different classes should not be allocated to one lecturer at same time.
4. There should be sufficient university resources for conducting the sessions as lecture halls and lecturers.

There are various causes for choosing the particular case as a target. The first reason is that the scheduler had already encountered constraints assumed for designing human made timetabling in real-time. The second reason will be the scheduler will attempt for comparison. The third cause is that

the case specified is absolutely not easy to solve computationally thereby enforcing implementation of more promising and highly efficient timetabling optimization technique. The parameters of the research case-study are:

- Inclusion of two categories of sessions e.g. laboratory and theory sessions with different attributes and restraints.
- The scheduling to be resolved has only 5 class rooms (lecture halls) which is challenging. Out of 5 class rooms, 3 rooms will be considered for conducting theory sessions and 2 for laboratory.
- Particular session should be delivered in particular class room, e.g. in theory class room, only theory can be conducted posing no chance of laboratory session.
- Another prominent situation is number of lecturers. Each branch has 60 students, so 180 students/semester as we have considered three branches (Computer Science, Information Technology, and Electronics & Communication). Similarly, there are total 60 lecturers for all branches. To understand the complexity of the problem under discussion, we can say that total of 200 time slots (8 Hours - 1 Hour per Class x 5 Days x 5 Rooms) has to be accommodated in 5 class rooms out of which we have only three rooms for conducting theory and two for laboratory sessions. The problem is complex for normal search algorithm. Hence, proven Genetic Algorithm is chosen to accomplish the best optimized result.

The proposed system using Genetic Algorithm initiates with the design of a population of feasible solutions to the timetabling problem. The algorithm-1 is designed with all considerations for creating an input file with the datasets provided. Algorithm-2 uses Genetic Algorithm for optimized result. Here each entity in the population will represent a chromosome and every chromosome will consist of genes to be grouped to represent a timetable. The cumulative number of the genes in the chromosome will again depend on number of rooms required to cover up all the branches as well as all the semesters of one branch with the same available resources, thereby striking a challenging environment.

$$W = \sum_{i=1}^{20} L_i + \sum_{j=1}^5 C_j + \sum_{k=1}^2 C_k + \sum_{l=1}^{60} S_l + \sum_{m=1}^8 Sem_m \quad (1)$$

Fitness Function is given by,

$$W / (total\_classes * 5) \quad (2)$$

As shown in equation (1), W is the weight factor which is dependent on number of lecturers (L), number of class rooms with and without laboratory (C), number of students (S), and number of semesters (Sem). Fitness function is shown in equation (2). It should be noted that the entire fitness function is designed to check the optimized result for one week only.

#### Algorithm-1: Constraints Design

**Input:** All the constraints parameters

**Output:** Generation of input file for optimization

**Steps:**

- 1 Design simulation parameters
- 2 Create a class for courses

3 Create a class for Lecturers

4 Create a class for Lecture halls

5 Create a class for Subjects

6 Create a class for Time slots

7 Create chromosomes (individual)

8 Initialize no. of crossover points of parent's class tuple

9 Initialize no. of classes moved randomly by single mutation operation

10 Estimate probability of crossover

11 Estimate probability of mutation

12 Design Fitness values of chromosomes

13 Flag the constraints

14 Design time-space slots

15 Design class table for chromosomes

16 Compute first time-space slots used by class

17 Create configuration input file

#### Algorithm-2: Genetic Algorithm Implementation

**Input:** Configuration input file

**Output:** Optimized Timetable

Initialize new population with chromosomes randomly built.  
Generation=0.

#### Repeat

Select parents for crossover Replace\_By\_Generation times.

Produce the same number of offsprings.

Apply mutation to all.

Select Replace\_By\_Generation chromosomes and replace them with offsprings by protecting best chromosomes.

Try to add new chromosomes in best chromosomes group.

Generation=Generation+1.

Until an algorithm has reached the criteria.

The arbitrary course is derived from the set of course tags and evaluated if it requires consecutive time instants, then allocate an appropriate time period considering no overlapping of sessions and involving one lecturer as well as lecture hall from their equivalent register for that course. The fitness function shown in equation (2) is designed considering hard constraints. The weight factor has consideration of all the hard constraints (lecturers, lecture halls, students, subjects and semesters) (See equation (1)). The estimation of fitness function will not be done overlooking any particular constraint. With the assistance of the selection of better solutions, the optimal section of the population is considered based on their fitness values.

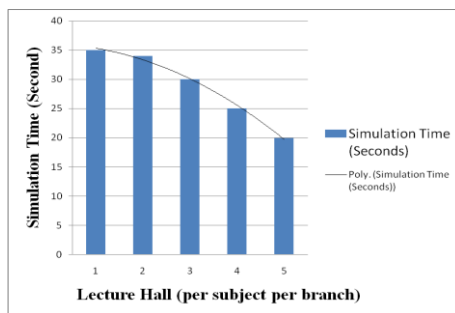
## 4. RESULTS and ANALYSIS

For the experimental analysis of the proposed curriculum timetabling system, a real time dataset of Dr. Babasaheb Ambedkar Technological University (Maharashtra, India) is considered. Three branches i.e. Computer Science, Information Technology and Electronics & Telecommunication are considered. The problem involves scheduling of total of 60 lecturers for 180 students in only 5 class rooms (lecture halls) for 5 working days in a week with 8 hours/day and 1 hour of lunch break. The proposed algorithm is implemented in Microsoft Visual Studio 2008 environment using VC++ programming language. The optimal system configuration is 32 bit Windows OS with minimum 1.84GHz dual core processor. The user interface is shown in Figure 1. The first set of the performance analysis is shown in Figure 2, with respect to the lecture hall (per subject

per branch) and simulation time to understand the run efficiency of the proposed algorithm to give the optimized results. It's clear that the algorithm should need much processing time when the lecture halls are less in number as the population will be highly iterated to check the violations of constraints and maximize the fitness function.

|         | MON                       | TUE                          | WED                    | THU                      | FRI                    |
|---------|---------------------------|------------------------------|------------------------|--------------------------|------------------------|
| 0 - 10  | Software Engineering (L1) | Data Structures Using C (L1) | Computer Graphics (L1) | Operations Research (L1) | Scheduling Theory (L1) |
| 10 - 11 | Software Engineering (L2) | Data Structures Using C (L2) | Computer Graphics (L2) | Operations Research (L2) | Scheduling Theory (L2) |
| 11 - 12 | Software Engineering (L3) | Data Structures Using C (L3) | Computer Graphics (L3) | Operations Research (L3) | Scheduling Theory (L3) |
| 12 - 13 | Software Engineering (L4) | Data Structures Using C (L4) | Computer Graphics (L4) | Operations Research (L4) | Scheduling Theory (L4) |
| 14 - 15 | Software Engineering (L5) | Data Structures Using C (L5) | Computer Graphics (L5) | Operations Research (L5) | Scheduling Theory (L5) |
| 15 - 16 | Software Engineering (L6) | Data Structures Using C (L6) | Computer Graphics (L6) | Operations Research (L6) | Scheduling Theory (L6) |
| 16 - 17 | Software Engineering (L7) | Data Structures Using C (L7) | Computer Graphics (L7) | Operations Research (L7) | Scheduling Theory (L7) |

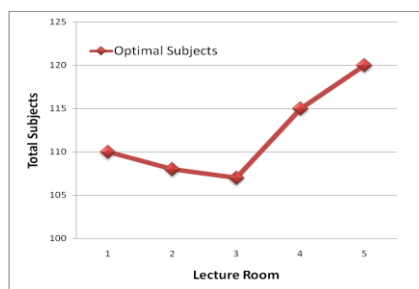
**Figure 1.**User interface of the proposed system



**Figure 2** Lecture hall vs Simulation Time

But the proposed algorithm takes maximum of some seconds to process the final result in much challenging environment i.e. of one room.

The second set of the performance analysis (Figure 3) is to understand the extent of optimization with respect to allocation of 5 available rooms for covering up all the 120 subjects (5 subjects x 8 semesters x 3 branches), without overlapping the classes. This is the main check for the hard constraints. It can be understood from Figure 3 that the algorithm should not take much time for 5 class rooms, but interestingly, it can be noticed that the algorithm is even successful in allocation of maximum of average 110 subjects in one week considering only 1 lecture hall per branch.



**Figure 3** Lecture Hall vs. Total Subjects

## 5. CONCLUSIONS

The proposed system presents an approach of solving the curriculum timetabling problem which is depicted as NP hard problem. The approach of Genetic Algorithm is deployed to estimate the best possible result in the real time dataset. The application can be readily scalable to the complex timetabling problem. This would be achieved by increasing the size of the University and by incorporating further constraints on the timetable. Genetic Algorithm effectively demonstrated an ability to solve complex optimization problem. Notably, this served to provide a very thorough introduction to the techniques employed and incorporated by Genetic Algorithm. The proposed system has well designed consideration of hard constraints towards scheduling. The scope of the proposed research work is multifold. With the rise of inclusion of various technical workshops as well as introduction of finishing schools within the premises of university, the university will definitely encounter extreme difficulty to manage it in same infrastructure. One probable solution will be either to cancel regular sessions and conduct the workshops or else university has to invest huge amount of money to build new blocks of buildings to conduct classes. The question is, is it possible to schedule the same events in existing infrastructure of the University? So, for the challenge, the proposed system has considered the problem of scheduling of 180 students with 60 lecturers and whopping 120 subjects to be taught in only 5 lecture halls. The problem definition has been restricted for only 1 week considering 5 working days. Extensive analysis and experiments were conducted in the designed prototype to see the better results satisfying the constraints.

But we also know that 100% optimization of available resources is not feasible in any case. For an example, the current work is unable to optimize lecturer for conducting maximum number of sessions in one week timeslots. The reason is that proposed algorithm basically works as fairness protocol to provide equilibrium and optimized result considering all the resources with fast accomplishment of final result. Reducing the lecturers will not solve the problem as according to optimization theory, reducing number of lecturers will mean more allocation of sessions for them, which will again contradict the theorem. However, our future direction will be to solve this case of optimization.

## 6. REFERENCES

- [1] Leonardo Aparecido Cicon, Humberto Cesar Brandão de Oliveira, Michelle Cristina Alves Andrade, Guilherme Bastos Alvarenga, Ahmed Ali Abdalla Esmin, "The School Timetabling Problem: a focus on elimination of open periods and isolated classes", Hybrid Intelligent Systems, HIS'06, Sixth International Conference, Dec. 2006.
- [2] A. Schaerf, "A Survey of Automated Timetabling", Journal Artificial Intelligence, Vol. 13, Issue 2, April 1999.
- [3] Dilip Datta, Kalyanmoy Deb, Carlos M. Fonseca, "Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm", in NCRSME 2007 -

- National Conference of Research Scholars in Mechanical Engineering, (Kanpur, India), Mar. 2007.
- [4] Andrea Schaerf, “Multi-Agent Systems for Optimization Problems: the University Course Timetabling Case Study”, WOA, Rende (CS), Italy, July 5th, 2011.
- [5] A. Chaudhuri, D. Kajal, “Fuzzy Genetic Heuristic for University Course Timetable Problem”, *Advance. Soft Computing Applications*, Vol. 2, No. 1, March, 2010, ISSN 2074-8523; 2010.
- [6] Anmar Abuhamdah, “Experimental Result of Late Acceptance Randomized Descent Algorithm for Solving Course Timetabling Problems”, *IJCSNS- International Journal of Computer Science and Network Security*, Vol.10 No.1, January 2010.
- [7] Burke, E. K., Bykov, Y., Newall, J., Petrovic, S., “A Time-Predefined Approach to Course Timetabling”, *Yugoslav Journal of Operation Research*, pp. 139-151., 2003.
- [8] Edmund Burke, Adam Eckersley, Barry McCollum, Sanja Petrovic, Rong Qu, “Using Simulated Annealing to study behaviour of various Exam Timetabling data sets”, *The Fifth Metaheuristics International Conference, MIC*, 2003.
- [9] Luca Di Gaspero, Andrea Schaerf, “Tabu Search Techniques for Examination Timetabling”, *Proceeding PATAT '00, Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling- III*, 2001.
- [10] E. Burke, K. Jackson, J. H. Kingston, R. Weare, “Automated University Timetabling: The State of the Art”, *Oxford Mathematics & Physical Sciences Computer Journal*, Vol. 40, Issue 9, 565-571.
- [11] URL: <http://patat2012.com/information.html>, Accessed on Nov-15, 2011.
- [12] URL: <http://www.utwente.nl/en>, Accessed on Nov-15, 2011.
- [13] Ruibin Bai, Jacek Blazewicz, Edmund K Burke, Graham Kendall, Barry McCollum, “A Simulated Annealing Hyper-heuristic Methodology for Flexible Decision Support”, *Computer Science Technical Report*, 2007.
- [14] Tunçhan Cura, “Timetabling Of Faculty Lectures Using Simulated Annealing Algorithm”, *Istanbul University*, 2007.
- [15] M. Nandhini , Dr. S. Kanmani, “A Survey of Simulated Annealing Methodology for University Course Timetabling”, *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, May 2009.
- [16] Baki Koyuncu, Mahmut Seçir, “Student Time Table by Using Graph Coloring Algorithm”, retrieved from [http://www.emo.org.tr/ekler/76e76856c7fea3b\\_ek.pdf](http://www.emo.org.tr/ekler/76e76856c7fea3b_ek.pdf), 2007.
- [17] Nasser R. Sabar, Masri Ayob, Graham Kendall, Rong Qu, “Roulette wheel Graph Colouring for Solving Examination Timetabling Problems”, *COCOA- 2009*, 463-470.
- [18] Dr. Cauvery N. K., “Timetable Scheduling using Graph Coloring”, *International Journal of P2P Network Trends and Technology*, Vol.1, Issue2- 2011.
- [19] Ho Sheau Fen, Safaai-Deris, Siti Zaiton-Mohd Hashim, “Investigating Constraint-Based Reasoning for University Timetabling Problem”, *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, Vol. 1, IMECS, March 18 - 20, 2009.
- [20] Ender Özcan, Alpay Alkan, “A Memetic Algorithm for Solving a Timetabling Problem: An Incremental Strategy”, *MISTA* 2007.
- [21] Nguyen Quoc Viet Hung, Ta Quang Binh, Duong Tuan Anh, “A Memetic Algorithm for Timetabling”, *Research Informatics Vietnam-Francophony*, Can Tho, Vietnam, 289 – 294, February, 2005.
- [22] Luca Di Gaspero, Andrea Schaerf, “Tabu search techniques for examination timetabling”. In E. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling*, Vol. 2079, *Lecture Notes in Computer Science*, pages 104-117. Springer Verlag, Berlin-Heidelberg, Germany, 2001.
- [23] Jean-François Cordeau, Brigitte Jaumard, Rodrigo Morales, “Efficient Timetabling Solution with Tabu Search”, *International Timetable Competition*, 2003.
- [24] Ben Paechter, “A Conflict Tabu Search Evolutionary Algorithm for Solving Constraint Satisfaction Problems”. In *Proceedings: EvoCOP*, 2008.
- [25] Aldy Gunawan, Kien Ming Ng, “Solving the Teacher Assignment Problem by Two Metaheuristics”, *International Journal of Information and Management Sciences* (2011), 73-86.
- [26] Kimmo Nurmi, Jari Kyngäs, “A Framework for School Timetabling Problem”, *MISTA- 2007*.
- [27] P. Cote, T. Wong and R. Sabouri, “Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem”, In: E.K. Burke and M. Trick (eds.) ,*Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling*, Springer, *Lecture Notes in Computer Science*, Vol. 3616, 151-168.
- [28] K. Sheibani, “An evolutionary approach for the examination timetabling problems”, In: E.K. Burke and P. De Causmaecker (eds.), *Proceedings of the 4<sup>th</sup> International Conference on Practice and Theory of Automated Timetabling*, August, 2002, Belgium, 387-396.
- [29] W. Erben, “A grouping Genetic Algorithm for graph colouring and exam timetabling”, *PATAT: Selected Papers from the 3<sup>rd</sup> International Conference*, Springer, *Lecture Notes in Computer Science*, Vol. 2079, 132-156.