www.ijecs.in
***International Journal Of Engineering And Computer Science ISSN:2319-7242***
***Volume 3 Issue 6 June, 2014 Page No. 6409-6413***

# Intelligent Course Timetable Scheduling Using Memetics

***Pooja Ghadiali[1], Prachi Mehta[1],Sachin Nagda[1], Dharit Shah[1], Prof. Purva Raut[2], Prof. Anuja Nagare[2]***

[1]B.E. Student, Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering,
University of Mumbai, Mumbai, India
*Email: 1) poojaghadiali@gmail.com,*
*2) mehta.prachi06@gmail.com,*
*3) sachinnn92@gmail.com,*
*4) shahdharitp@gmail.com,*

[2]Assistant Professor, Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering,
University of Mumbai, Mumbai, India
*Email:1)purvapraut@gmail.com,*
*2) nagare.anuja@gmail.com*

**Abstract:*****A college timetable is a temporal arrangement of a set of lectures and classroomsin which all given constraints are satisfied. Hence, timetabling problem is one of the best examples of a Constraint Satisfaction Problem (CSP). Memetic Algorithm is one of the best techniques to solve the Constraint Satisfaction Problem(CSP) and it is a combination of standard genetic algorithm and hill climbing algorithm. In this paper Course Timetable Scheduling System (CTSS) is developed to solve Timetabling – Constraint Satisfaction Problemwhich makes use of Memetic algorithm.***

**Keywords:** Course Timetable Scheduling System (CTSS), Constraint Satisfaction Problem (CSP), Hard Constraints, Fitness Function, Memetic Algorithm**.**

## 1. Introduction

A college timetable is a temporal arrangement of a set of lectures and classrooms in which all given constraints are satisfied. Hence, timetabling problem is one of the best examples of a Constraint Satisfaction Problem (CSP). University course timetabling problems are a subclass of course timetabling problems that require feasible assignment of each offered course to a slot of a discrete timetable and some other resources such as classrooms, satisfying a set of constraints. The timetabling problem also arises in various other forms such as nurse scheduling and transportation timetabling. It is defined with the help of the following parameters:

- Time slots
- Set of resources (e.g. classroom / teachers in case of course timetables)
- Set of constraints namely hard constraints and soft constraints

The problem is to allot timeslots and resources while seeing that all the necessary constraints are satisfied. As mentioned above, there are two types of constraint namely, hard and soft. While hard constraints are mandatory to be satisfied, it is not necessary to satisfy all the soft constraints, although it would be optimal to do so. The solution may be accepted even if soft constraints are violated. This paper presents a feasible implementation of time table generating system for anEducational Institution. The principle objective of the system is generating a system which reduces the work load on the human while generating a feasible time table for the various departments of anEducational Institution.

## 2. Course Timetabling

Burke, Kingston and de Werra[1] gave a definition of general timetabling as a problem with four parameters: given T, a finite set of times; R, a finite set of resources; M, a finite set of meetings; and C, a finite set of constraints, the timetabling problem is to allocate times and resources to the meetings such that the constraints are satisfied. Timetablingproblems are is an NP-hard problem that requiresto satisfy all the hard constraints.

There are various approaches to generate a feasible timetable. One of them can be specified as assigning a room to time-events slots and assigning teachers to subjects and subjects to rooms.

Data is entered for variety of parameters such as the list of rooms available, the time slots, subjects and names of teachers. After this assignment procedure is completed, memes are generated followed which all the resources of the university are assigned such that all the hard constraints are satisfied.

There are two types of constraints: hard and soft. Hard constraints are those constraints which are mandatory to satisfy whereas soft constraints are those constraints which generate feasible timetables even if they are not satisfied. A feasible timetable is one which does not violate any hard constraints and satisfy maximum soft constraints. The following hard constraints [2] are considered during the course of this software:

- h1: No classroom/lab is allotted to more than 1 division at a particular time (room conflict).
- h2: A particular professor should be teaching in only 1 class at a time (professor conflict)
- h3: At a time, 1 division is supposed to be in 1 classroom/lab (no student's time conflict)
- h4: Every professor should have at least 1 lecture everyday i.e. he should not be completely free on any day.

Following are few of the soft constraints[2] used in the implementation:

- s1: Break is given in a particular time slot
- s2: Teacher can select his/her choice of timings (teacher time preference)
- s3: load should be evenly distributed throughout the week for a teacher

## 3. Memetic Algorithm

Evolutionary algorithms are alone not enough to generate optimized timetables. Hence, memetic algorithm is used because it combines evolutionary algorithm framework with problem specific local search heuristics. A meme, a cultural unit of evolution, represents a unit of data. Once a meme is created, data within the meme cannot be changed (similar to a chromosome). In addition, the concept of meme is taken to represent learning or development strategies that are employed to improve individuals, in this case, improve timetables. The process of improving timetables is done with the help of a local search based on a hill climbing algorithm. This will exploit the best candidate solutions while the genetic algorithm serves as an outer frame for exploring the search space for the local search algorithm. The detailed algorithm is given in the later section.

### 3.1 Solution Representation

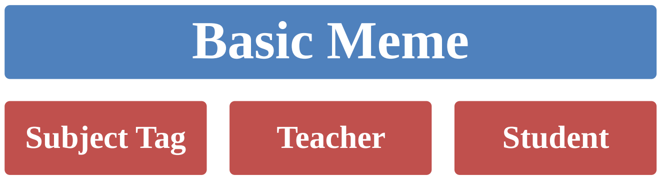

**Figure 1:**Basic Meme Structure

Figure 1 shows the Basic Meme Structure. A meme is discrete package of culture. In this case the meme is a collection of information which can be used as a single data structure. It is always used to convey some information. It is analogous to a gene. Once the meme is generated, two operations can be performed: selection and crossover. It is done in order to provide a more optimal result. In this case when the user enters data in the teacher manager, subject manager and student manager, the information is saved in an xml file. After which, the user needs to create a link between student and subject as well as teacher and subject. This helps the software to know what teacher teaches what subject and which subject is taught to which year (FE/SE/TE/BE).When one generates activities, the memes are generated. These are then placed as per the various constraints that need to be satisfied using the genetic algorithm to create the time table.

### 3.2 Fitness Function and Fitness Distance

Fitness Function is specified in Equation 1[1]. Fitness Function also known as cost function which is basically the score of a meme with its regard to its satisfaction of all the applied hard constraints. If it satisfies a hard constraint, its score is incremented for that constraint. The total fitness function for a complete timetable is Summation of all the individual meme's fitness. The solution is considered to be fit if all its meme's satisfy all the hard constraints and that solution is considered to be the most optimal if all its hard as well as soft constraints are satisfied. Fitness Distance is an entity in Equation 2 which is used to calculate how far the timetable is from being feasible where E is the total number of lectures, H is the total number of Hard Constraints and h is the hard constraint.

Fitness Function

$$f= \frac{\sum_{c=1}^{E}\sum_{i=1}^{H} h_{i,c}}{ExH} \qquad (1)$$

Fitness Distance

$$fd=1- \frac{\sum_{c=1}^{E}\sum_{i=1}^{H} h_{i,c}}{ExH} \qquad (2)$$

### 3.3 Genetic Operator

After completion of generation of memes following genetic operations[2] are carried out:

#### 3.3.1 Selection Operator

Selection is the stage of a genetic algorithm in which individual meme's are chosen from a population for later breeding. After creation of the Meme's all the meme's are stored in a vector in the ascending order of their primary ID. Meme'sare stored in a vector in random order. After this they are selected one by one in order to perform further operations.

#### 3.3.2 Partially Matched Crossover (PMX) operator

Crossover is a genetic operator that combines two parent meme to produce a new offspring. The idea behind crossover is that the new meme may be better than both of the parents if it takes the best characteristics from each of the parents. In this method, two meme's are selected at random and PMX proceeds by position wise exchanges. The two crossover points give matching selection. It affects cross by position-by-position exchange operations. In this method parents are mapped to each other, hence we can also call it partially mapped crossover. In the given system, one variable is different for various meme's (for e.g. the duration). Whenmeme's with different durations are selected, this crossover function helps to tackle mapping problem easily.

### 3.3.3 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population to the next. Mutation alters one or more meme values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Mutation occurs during evolution according to a user-defined mutation probability. Over here the system searches for the meme's which conflict the constraints and also searches for the pair of such meme's. As soon as the pairs are spotted it undergoes Mutation on those selected meme's and forms a new solution. It uses Neighborhood functions of MoveRoom and MoveTime so as to perform Mutation without violating any constraints. Once the Mutation is performed it checks the fitness value of these meme's and accordingly performs the next step.

### 3.4 Neighborhood Function

For the complete formation of the timetable few methods called as neighborhood functions [1] are used in using which few moves are implemented, for e.g. : MoveTimeSlot, MoveClass and IsActivityPlaced. The first method basically reschedules a given activity from one time slot to another depending upon the constraints it satisfies and the other method MoveClass replaces the classroom of the desired lecture of any course without changing the timeslot. Again all the constraints are looked upon. The IsActivityPlaced method is basically a Boolean method which returns true if the particular meme is placed in the timetable or not and returns false otherwise.

### 3.5 Memetic Algorithm

```
In the following algorithm, following variables are assigned
N: Total number of memes
i = 0
1. Generate all the meme’s.
2. Generate a random sequence of the Meme’s.
3. For i=0 to i=N; i++
4. Start placing the meme’s using selection operator.
5. Check the conflicting meme’s.
6. Select ‘n’ pairs of conflicting meme’s and perform mutation.
7. Calculate the fitness distance (fd),
8. If (fd != 0) then perform Neighborhood functions i.e. MoveRoom, MoveTime methods.
9. Else Proceed to next step.
10. Perform Partial Matched Crossover to find the best possible solution.
11. Store the Meme’s in the matrix representation.
12. If (fd∀ N Meme’s != 0) then go to step 3.
13. Print the timetable in all the views specified.
```

## 4. Architecture of the System

Figure 2 shows Architecture of the system. The architecture of the system is divided into 4 levels. The first level is used for acquiring following input from the user: Student data, Teacher data, Subject data, Room data and Timeslot data.

The second level is used for linking student data with that of the teacher data and teacher data with that of the subject data.

Once data is linked, memes are generated.From the generated list of memes, genetic operations are performed in the third level.

The last level is exporting the generated timetable as a .pdf or a .html file.

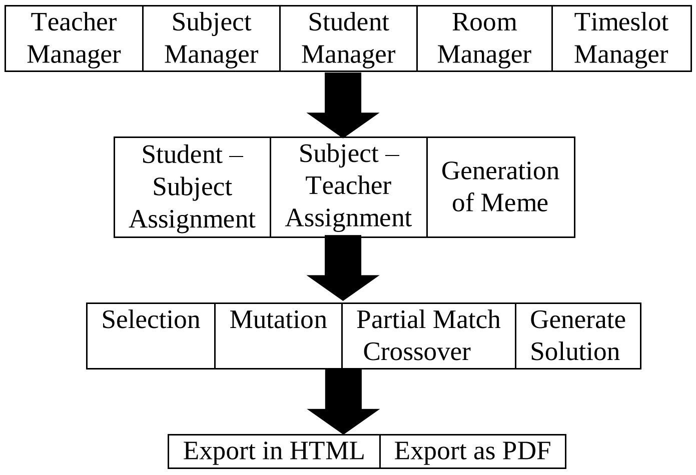

**Figure 2:**Architecture of the System

## 5. Features of Our System

With the help of the following example we can explain the various features of our system. Let there be two teachers T1 and T2, two rooms: R1& R2, two labs: L1& L2 (which have a maximum capacity of 25 students) and four years of students: S1, S2, S3, and S4.

1) Room or lab is not assigned to the two different activities at the same time. If S4 is to have a lecture in room R1, then the software knows that room number R1 is occupied and cannot be assigned to S1 or S2 or S3 who may have lectures at the same time.
2) Teacher is not assigned to more than one activity at a time. If T1 has a lecture from 9.30 to 10.30 then T1 is not available to others during that slot.
3) Students are not assigned to more than one activity at a time.S1 cannot have a lab and lecture scheduled at the same time.
4) Room or Lab for an activity should have enough capacity to fit the number of students. All the batches of S3 cannot have class in the L1 lab at the same time as the capacity is only 25 students; this is taken care of by the system.
5) Break time constraint can be set.The user is given a drop down from which he can select at what time he wants to set the break.
6) Maximum number of hours per day can be set.The maximum number of hours per day and number of days of college are taken as an input from the user as this varies in various colleges.
7) If a student is not available at a particular time or on a particular day, it is possible to satisfy that constraint. For example S4 students should have one project day, so they cannot have any lecture or practical session on that day. A particular day needs to be kept free. The user is given awhile setting the rules.
8) If a teacher is not free at a particular time or on a particular day, it is possible to satisfy that constraint.If an institute has a visiting faculty who may not be available on all days, it is possible to accommodate this constraint.
9) If a room is not available at a particular time or on a

particular day, it is possible to satisfy that constraint.

10) Practical Session can be taken by multiple teachers depending on the availability.For example T1 and T2 can take the same subjects practical for two different batches of the same class.

It is possible to generate the time table again if the timetable is not as optimal as the user wants. This time table will also satisfy all the constraints mentioned above.

## 6. Comparison with Exciting System

Figure 3 and 4 show the comparison of the CTSS software with the existing system. Comparison is based on the time required to generate the time table and the number of constraints satisfied by the system

### 6.1 Time Requirement

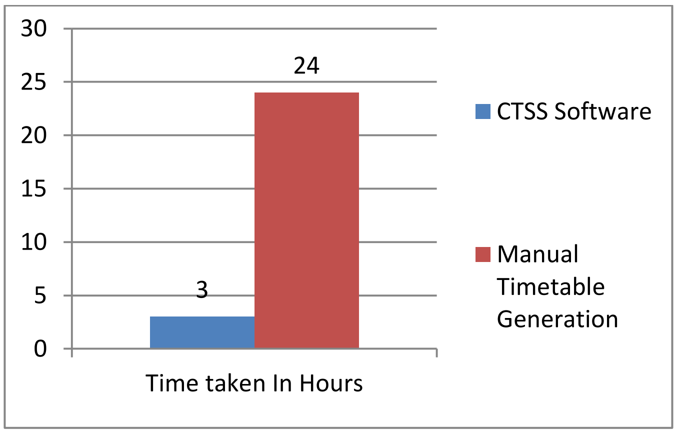

**Figure 3:**Time comparison graph of CTSS software and manual Time table generation

As per the survey done in our institution, the generation of a timetable takes up to 24 hours. Whereas if the timetable generation software is used, it takes up to 3 hours, right from entering all the relevant data till exporting the approved timetable.
If the generated timetable is not approved by the faculty or the institute, a new timetable can be generated in a span of 5 minutes which isn't the case in the manual process.

### 6.2 Numberof constraints satisfied by the system:

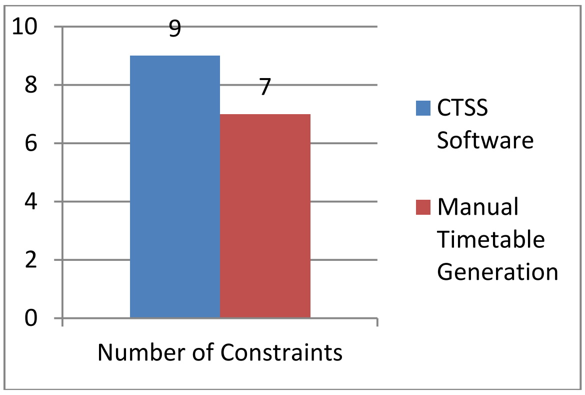

**Figure 4:**Number of Constraints Graph

The accuracy of timetable is decided on the basis of the number of soft constraints it satisfies. As seen from the figure 4, CTSS software satisfies more soft constraints which are not considered while designing a timetable using the manual system.

## 7. Conclusion

The existing manual system is time consuming and tedious. This project has been developed to overcome these limitations and ease the process of time table generation. Variousconstraints that apply to our institution time table generating system are considered while developing this software.

The system to generate a timetable is completed successfully by in using Java as front end and XML as backend. Separate timetable for students, rooms, teachers and batches of students are generated automatically by this system. An optimal timetable is generated by this software using the memetics algorithm. The system reduces the time consumption and the stress of framing it manually. System has a simple UI which makes it easy for the user to enter the data and generates the timetable as per the rules selected by the user. It is possible to alter the hours, days and all the other available resources such that it can be used to generate a time table for any educational institution

## References

[1] Tri A. Budiono and KokWai Wong, "Course Timetabling using Memetics" Murdoch University, Western Australia,

[2] O.Rossi-Doria, C.Blum, J. Knowles, M. Samples, K. Socha, B. Paechter,"A Local Search for Timetabling Problem," in Proceedings of the 4th Internatiomnal Conference on Practice and Theory of AutomatedTimetabling 2002, Gent, Belgium, August 21-23

[3] Ashish Jain1, Dr. Suresh Jain2 and Dr. P.K. Chande, "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable", International Journal of Innovation, Management and Technology, Vol. 1, No. 3, ISSN: 2010-0248, August 2010

[4] AlpayAlkan Ender Özcan, "Memetic Algorithms for Timetabling", Yeditepe University, Department of Computer Engineering

[5] P. Moscato, "Memetic algorithms: A short introduction," in New Idea in Optimization, D. Corne, F. Glover, and M. Dorigo, Eds. New York : McGraw-Hill, 1999, pp. 219-234

[6] L. Merlot, N. Boland, B. Hughes, and P. Stuckey, "A hybrid algorithmfor the examination timetabling problem," in The Practice and Theory ofAutomated Timetabling (PATAT IV). vol. 2740 Berlin: Springer, 2004

[7] A. S A. Schaerf, "A survey of automated timetabling," Artificial IntelligenceReview, vol. 13, pp. 87-127, 1999

[8] M., Jankovic, "Making a Class Schedule Using a GeneticAlgorithm,"Available: http://www.codeproject.com/KB/recipes/GaClassSchedule.aspx.[Accessed: April May 3, 2011], (2008)

[9] T. B. Cooper and J. H. Kingston, "The complexity of timetableconstruction problems," in Proceedings of

Practice and Theory ofAutomated Timetabling (PATAT-95), E. K. Burke and P. Ross, Eds.Berlin-Heidelberg: Springer-Verlag, 1996, pp. 283-295

[10] Tri A. Budiono, KokWai Wong, "Memetic AlogrithmBehaviour on Timetabling Infeasibility" , School of Information Technology, Western Australia

[11] E. Burke, J. Kingston and D. de Werra, Applications to Timetabling, Handbook of Graph Theory, (J. Gross and J. Yellen eds.), pp. 445-474, (Chapman Hall/CRC Press, 2003).

## Author Profile

**PoojaGhadiali** is pursuing B.E. inInformation Technology Engineering from Dwarkadas J. Sanghvi College of Engineering.

**Prachi Mehta**is pursuing B.E. in Information Technology Engineering from Dwarkadas J. Sanghvi College of Engineering.

**SachinNagda**is pursuing B.E. in Information Technology Engineering from Dwarkadas J. Sanghvi College of Engineering.

**DharitShah**is pursuing B.E. in Information Technology Engineering from Dwarkadas J. Sanghvi College of Engineering.

**Prof. PurvaRaut** received the B.E. in Computer Engineering from Mumbai University and M.Tech. in Computer Engineering from NMIMS University in 2005 and 2008, respectively. She is now working as Assistant Professor in Department of Information Technology in Dwarkadas J. Sanghvi College of Engineering.

**Prof. Anuja Nagare** received the B.E.in Computer Engineering from Dwarkadas J. Sanghvi College of Engineering and M.E. in Computer Engineering from ThadomalShahaniEnginerring College in 2009 and 2012, respectively. She is now working as Assistant Professor in Department of Information Technology in Dwarkadas J. Sanghvi College of Engineering.