International Journal of Research
eISSN: 2348-6848 & *pISSN:* 2348-795X *Vol-5 Special Issue-13*
International Conference on Innovation and Research in Engineering, Science & Technology
Held on 23rd & 24th February 2018, Organized by Tulsiramji Gaikwad Patil College of Engineering & Technology, Nagpur, 441108, Maharastra, India.

Automatic Class Timetable Generation using a Hybrid Genetic and Tabu Algorithm

Prof. Premlata B. Shahare
Dept. of Computer Science and Engineering,
TGPCET, Nagpur
Email: premlata.sonawane@gmail.com

Prof. Priyanka Bhende
Dept. of Computer Science and Engineering
TGPCET, Nagpur
Email: bhendepriyanka26@gmail.com

Abstract – Timetable generation is a combinatorial optimization problem. Meta Heuristic methods and Evolutionary Algorithms have given the best results when it comes to solving the problem of timetable generation. In this paper the problem of timetable generation for the Computer Genetic Algorithms help in finding multiple optimal solutions in one iteration but they can get stuck at local optima. This can be avoided by using Tabu Search procedure.
Key Words - Course timetabling, Genetic Algorithm, Tabu Search.

I. INTRODUCTION

Timetable generation is a very intriguing problem belonging to the class of NP- hard problems. Timetable generation is nothing but the allocation of subjects, lecturers and students into appropriate timeslots and classrooms in such a way that they satisfy a predefined set of constraints. Since timetabling is a NP-hard problem, it is not possible to solve it using conventional optimization techniques like backtracking or constraint logic programming as they fail to generate optimum solutions [1]. In recent times a lot of research has been done in the field of Evolutionary computation algorithms and Meta- Heuristics. Genetic Algorithms and Tabu Search which belong to the above mentioned methods are the most preferred ways of solving the University course timetablingproblem as it not only eliminates the violation of the essential constraints but also ensures reaching global maxima.

II. PROBLEM DESCRIPTION

The timetables considered in this paper timetables are generated for the courses that belong either to the odd semester group or the even semester group (i.e., Semester 3, Semester 5, Semester 7 or Semester 4, Semester 6, Semester 8). Classes are conducted on 6 days of the week i.e., from Monday to Saturday. The duration of each timeslot is 55 minutes. The number of lectures and labs to be allotted to a course depends on the credits assigned to that course.

III. PROBLEM DEFINITION

The course timetabling problem can be articulated asfollows:
P = {A, B, C, D, E} where A = {a1,a2,a3,...am} is the set of subjects, B = {b1,b2,...bn} is the set of teachers, C ={ c1, c2, c3, c4, ..c6} where each ci (for i = 1 to 6) represents the set of timeslots available during each day of the week. D = {d1, d2,, ..dt} is the set of classrooms and E =

{e1,e2,….ef} is the set of constraints. Constraints can be classified as:
Hard Constraints:
E1) One teacher cannot take two classes at the same time.
E2) A group of students cannot attend two classes at the same time.
E3) Two different classes cannot be held at the same room at the same time.
E4) There cannot be any classes beyond the schedule of the college.
Soft constraints:
E5) Make sure that a group of students don't have to come to college only to attend one class.
E6) Not more than two continuous slots should be allocated for a particular subject.
E7) No class must be allotted during break time.
E8) Two sessions of the same subject should not have a break between them.
E9) The weekly lecture hours must be within the maximum allowed hours.
E10) The number of classes allotted per week for a subject must be determined by its credits.
In our problem we use a combination of Genetic Algorithm and Tabu Search Technique.

IV. FLOW CHART

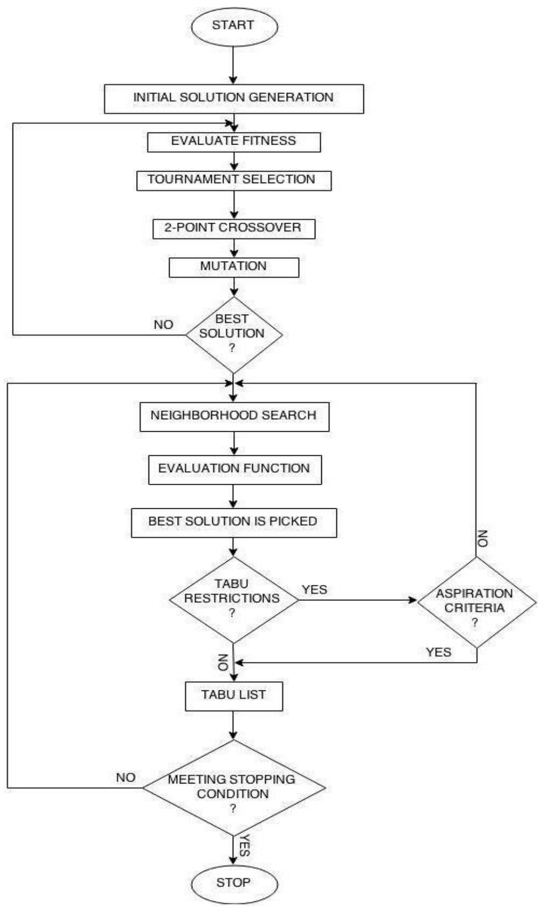


Figure 1. Proposed flow of the algorithm.

V. ALGORITHMS

A. Genetic Algorithm (GA)

The concept of Genetic Algorithms was presented by J.H. Holland. Genetic Algorithms imitate the process of natural evolution. They can be used to solve problems whose solution lies in large solution spaces. The advantage of Genetic Algorithms is that they reach global optima. To begin with, a set of initial solutions is generated. This set is called the population. Each population consists of a set of chromosomes. In this paper each chromosome represents a timetable. Each chromosome is made up of several genes. Here each gene is nothing but a course and its allotted time slot and classroom [21].These chromosomes are represented by bit strings. This is done to increase the efficiency of the algorithm. It also simplifies the process of mutation and crossover. An example of this binary encoding is shown in Table I.

Chromosome 1	0	1	1	0	0
Chromosome 2	1	1	1	0	1

TABLE I. BINARY REPRESENTATION OF CHROMOSOMES.

The next step is to optimize the initial solutions in the subsequent iterations [3]. Each chromosome has a parameter attached to it, which is the fitness. Fitness is a determinant of the number of constraints violated by each timetable [6]. Once the initial sets of solutions are generated, the fitness of each chromosome is calculated by the fitness function. Now, the fittest of the chromosomes are selected for generating the next generation of timetables. For this purpose the selection operatorof Genetic Algorithm is used. Here the concept of Evolution is used [7]. In the simulation of evolution, only the fittest individuals survive and the inferior individuals go extinct [20]. Thus only the fittest individuals in each generation survive. The evolution process is carried out by the selection, crossover and mutation operators. [2]. There is a small probability that the best solution can be eliminated during the selection process. In order to avoid this, a mechanism is devised to protect the best solution and ensure its presence in the gene pool. This is called elitism [2]. In this paper tournament selection is used as it ensures elitism. The next step is to perform crossover. Crossover provides structured yet randomized information exchange between individuals [4]. The crossover operator makes use of two individuals to create the next generation of off springs. Here, two point crossoversare used. Selection and crossover can lead to local optima. In order to avoid this, the mutation operator is used. The percentage of mutation is decided by the user. Mutation does the following things [5]:

1. Select an event which is causing collisions and assign it to a different slot.
2. Select and event randomly and assign it to a different time slot.
3. Select an event randomly and change only the room allotted to it.

The process of selection and crossover is used until a fixed number of generations are created and the optimal solution is obtained [8].

Algorithm 1: Genetic Algorithm ( )

```
Input: current generation a
Size of population b
Rate of elitism c
Rate of mutation d
Number of iterations e
Output: x
//Initial solutions
1. Generate b solutions randomly
2. Save them in the population p
3. for i = 0 to e do
//Selection on the basis of elitism
4. Number of elite chromosomes (f) = b*c
5. Select the best f solutions from p & store them in p1
//Crossover
6. Number of crossover (cr) = (b-f)/2
7. for j = 0 to cr do
8. Randomly select 2 solutions x1, x2 from p1
9. Generate x3, x4 by applying 2-point crossover to x1 &x2.
 10. Save x3, x4 to p2
11. End for
//mutation
12. for j = 0 to cr do
13. Select a solution xj from p2
14. Mutate xj under rate of d & generate new solution xj’
15. If xj’ is unfeasible
16. Update xj’ with a feasible solution by repairing xj’
17. End if
18. Update xj with xj’ in p2
19. End for
//updating
20. Update p = p1 +p2
```

21. End for
22. Return the best solution x in p.

B. Tabu Search:

Tabu search is one of the most popular local search methods based on the neighborhood search algorithm. Fred Glover proposed the Tabu search in 1986.Tabu search can be applied to any kind of optimization problem to get near optimal or optimal solution to the problem [17] . Thus the Tabu search would be one of the few ideal search techniques that is used to solve the combinatorial optimization problem of timetable scheduling. This is a kind of heuristic search technique that has the advantage of internal memory [15]. This internal memory is called the Tabu List. Tabu list is a First in First out list which contains the best solutions found so far. The major kind of moves used here are the neighborhood search moves [19].One of the main advantages of Tabu search is that it does not get stuck at local optima. This is achieved by allowing non-improving moves when it gets trapped at local optima. Non-improving moves directs the search away from local optima. The basic idea behind Tabu search is the 'Tabu', which means not-allowed. Tabus are used to prevent cycling of the same solutions whenmoving away from local optima [18]. But sometimes tabus are too powerful; they may prohibit useful moves even when there
is no danger of using them. Such moves also can be allowed using Tabu search if they satisfy a criterion called the 'Aspiration criteria' [10].
In order to obtain the best Timetable schedule for our university we produce a group of candidate timetable solutions from the initial timetable. The initial timetable solution is obtained by applying greedy techniques or from the timetable obtained from the Genetic Algorithm method. This initial timetable obtained using Genetic Algorithm method is not so fit and it is the fittest solution that can be obtained using Genetic Algorithm [16]. The quality of this solution can be improved greatly by using Tabu Search method. From this initial solution a group of candidate solutions is generated by applying neighborhood search techniques. We then pick the best solution from the group of candidate solutions. Then we again obtain a list of candidate timetable solutions from the best solution by again applying neighborhood search techniques [9]. This method is then repeated for a fixed number of iterations. At each iteration the best solution is added to the Tabu list andthis solution remains there for a certain number of iterations called the Tabu tenure. The Tabu tenure Tb , is the square root of the total number of courses in the timetable problem and its value is randomly chosen between 0.25 Tb to 0.50 Tb . The best of the new solutions is selected again and is tested for the Tabu list entry and the aspiration criteria [11]. They are put in the Tabu list if they pass the former test or if they satisfy the aspiration criteria. The Tabu search uses its memory ability to prevent cycling of the previously visited timetable solutions. Ateach iteration the best timetable solution is selected based on an evaluation function. This is a function of the soft constraint violation and the penalty incurred for the violation of each soft constraint is considered for calculation the value of this function. This is called the objective function. The main aim of the objective function is to minimize the cost of violation the soft constraints.
f = S Vi Wi. (1)
Where Wi is the penalty incurred for each soft constraint violation and Vi is the number of students involved in the soft constraint violation. Vi is equal to zero if no students are involved in the violation of the soft constraint.The major kind of moves used in the Tabu search is the neighborhood moves. These neighborhood moves are basically of the types N1, N2, N3 and N4. The neighborhood search is going to make swaps depending on the lengths of the lectures (some lectures are one hour long and some are two hour long) [12].
N1: This is a simple move where one lecture of a course is moved from the current period to a clash free period position.
N2: Here two lectures belonging to two different periods, rooms and days are swapped.

N3: Two different lectures in the same room but different periods are swapped.
N4: Two lectures that belong to different slots on the same day are swapped.
For each best solution obtained we define a neighborhood N(s) which has all the feasible solutions that are obtained by applying the neighborhood search method. Whenever a feasible solution has been reached we generate a subset V* of N(s) and we obtain the best solution s* in V* [13].

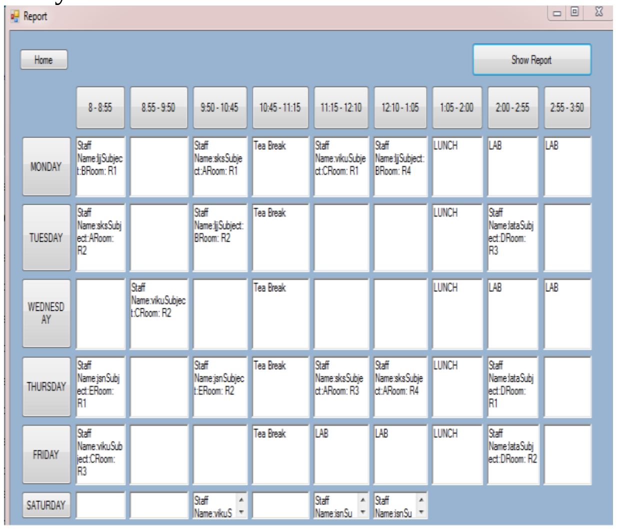
Algorithm 2: Tabu Search( )

```
1. Sol* = Sol;
2. f(Sol*) = f(Sol);
3. Set up Tabu List, TL;
4. I = Total number of iterations;
5. Set iteration I;
6. Do While ( i < I OR f(Sol*) = 0 )
7. Determine complete Neighborhood N (N1,N2, N3, N4) of current solution Sol;
8. Choose the best non-tabu solution Sol* form N;
9. Sol* = newly obtained Sol*;
10. Add Sol* to the Tabu List TL;
11. Update the best found solution;
12. End
13. Return the best found Solution.
```

The Tabu search process is stopped when the best solution found so far satisfies all the soft and hard constraints or the total number of iterations has reached say 3000 or if the total consecutive unimproved iterations is say 1000.
Thus the combination of Tabu search and Genetic Algorithm technique can be efficiently used for the automated scheduling of the timetabling problem which was previously done manually.

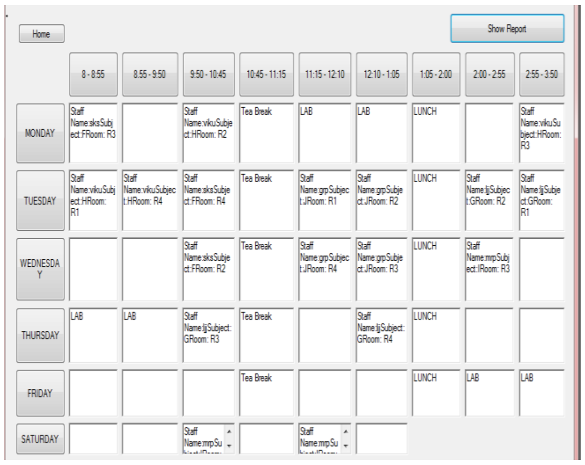
VI. IMPLEMENTATION

The implementation was done on Visual Studio 2012 and C# was used as the coding language. Also SQL Server Management Studio was used. Figures 2 and 3 represent the time tables generated using the methods mentioned in this paper. The time tables generated were found to satisfy all the hard constraints. The slots show teacher's name, subject and the room allotted. Timetables were generated for 2 sections of every semester under consideration.



	8 - 8:55	8.55 - 9:50	9:50 - 10:45	10:45 - 11:15	11:15 - 12:10	12:10 - 1:05	1:05 - 2:00	2:00 - 2:55	2:55 - 3:50
MONDAY	Staff Name:tjjSubject:BRoom: R1		Staff Name:sksSubject:ARoom: R1	Tea Break	Staff Name:vikuSubject:CRoom: R1	Staff Name:tjjSubject:BRoom: R4	LUNCH	LAB	LAB
TUESDAY	Staff Name:sksSubject:ARoom: R2		Staff Name:tjjSubject:BRoom: R2	Tea Break			LUNCH	Staff Name:lataSubject:DRoom: R3	
WEDNESDAY		Staff Name:vikuSubject:CRoom: R2		Tea Break			LUNCH	LAB	LAB
THURSDAY	Staff Name:jsnSubject:ERoom: R1		Staff Name:jsnSubject:ERoom: R2	Tea Break	Staff Name:sksSubject:ARoom: R3	Staff Name:sksSubject:ARoom: R4	LUNCH	Staff Name:lataSubject:DRoom: R1	
FRIDAY	Staff Name:vikuSubject:CRoom: R3			Tea Break	LAB	LAB	LUNCH	Staff Name:lataSubject:DRoom: R2	
SATURDAY			Staff Name:vikuS		Staff Name:jsnSu	Staff Name:jsnSu			

Figure 2. Timetable generated for semester IV.



	8 - 8:55	8.55 - 9:50	9:50 - 10:45	10:45 - 11:15	11:15 - 12:10	12:10 - 1:05	1:05 - 2:00	2:00 - 2:55	2:55 - 3:50
MONDAY	Staff Name:sksSubject:FRoom: R3		Staff Name:vikuSubject:HRoom: R2	Tea Break	LAB	LAB	LUNCH		Staff Name:vikuSubject:HRoom: R3
TUESDAY	Staff Name:vikuSubject:HRoom: R1	Staff Name:vikuSubject:HRoom: R4	Staff Name:sksSubject:FRoom: R4	Tea Break	Staff Name:grpSubject:JRoom: R1	Staff Name:grpSubject:JRoom: R2	LUNCH	Staff Name:tjjSubject:GRoom: R2	Staff Name:tjjSubject:GRoom: R1
WEDNESDAY			Staff Name:sksSubject:FRoom: R2	Tea Break	Staff Name:grpSubject:JRoom: R4	Staff Name:grpSubject:JRoom: R3	LUNCH	Staff Name:mrpSubject:IRoom: R3	
THURSDAY	LAB	LAB	Staff Name:tjjSubject:GRoom: R3	Tea Break		Staff Name:tjjSubject:GRoom: R4	LUNCH		
FRIDAY				Tea Break			LUNCH	LAB	LAB
SATURDAY			Staff Name:mrpSu		Staff Name:mrpSu				

Figure 3. Timetable generated for semester VI.

VII. EXPERIMENTAL RESULTS:

The below figure 4 shows the comparison of rate of convergence of Genetic Algorithm and Tabu Search for the Time Table Generation Problem. The initially generated solutions have theObjective value in the range of 150 to 450.

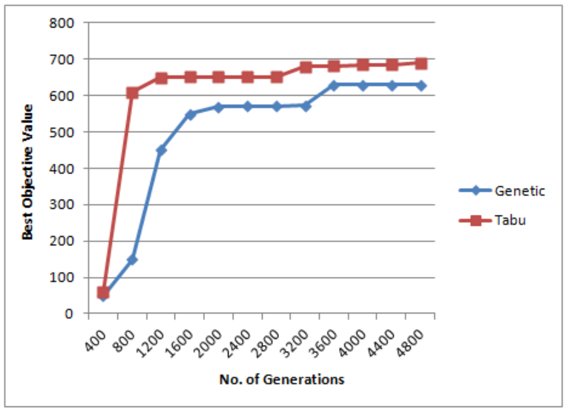


Figure 4. Convergence of Genetic Algorithm and Tabu Search.

VIII. FUTURE ENHANCEMENT

In the future we plan to make it more efficient by trying to satisfy all the mentioned soft constraints. This results in a timetable that is more optimum. Also, more soft constraints will be added to satisfy all the requirements. An option would be provided to increase the number of classrooms available so that timetables for more classes can be generated. An edit feature can be added so that a slot which causes inconvenience can be shifted to any other desired slot on a different day.

IX. CONCLUSION

Solving the problem of Timetable Scheduling using Genetic Algorithm has proven to give good results. But the disadvantage of this approach is that it gets stuck in local optima. The quality of the solution obtained using Genetic Algorithm can be highly improved by using a Hybrid Search technique that combines the qualities of Genetic Algorithm and Tabu Search approaches because Tabu search has the advantage of escaping local optima and it also achieves near global optimum solution. For the problem under considerationthe results obtained using this Hybrid Search technique are much better than the results obtained manually. This algorithm can also be used for the timetable scheduling of other universities by just modifying the constraints and requirements according to the particular university.

X. REFRENCES

[1] Sandeep Singh Rawat,Lakshmi Rajamani ,"A Timetable prediction fortechnical educational system using Genetic Algorithm",Journal of theoretical and Applied Information Technology,Vol.13(1), pp. 59-64, (2010)

[2] Ho Sheau Fen, Deris, Safaai, Mohd Hashim, Siti Zaiton,"Solving University Course Timetable Problem Using Hybrid Particle Swarm optimization",Conference on Intelligence and Human-Oriented Computing. LNCS, vol. 4733, pp. 848-855, (2009).

[3] Nabeel R. Al-Milli, "Hybrid Genetic Algorithms with simulated annealing for University course Timetabling Problem";Journal of Theoretical and Applied Information Technology,. Vol. 29 No.2, pp. 100-106, 31st July (2011).

[4] Arvind.S.Babu, R.Chockalingam, S.Kavitha, "A Hybrid Genetic Algorithm Approach to a Departmental Class Timetabling Problem Using E_cient Data Structures",IEEE , International Journal of Computer Applications (0975 - 8887) Vol. 1. No. 17, pp.99-103, (2010).

[5] Vibhor Bhatt, Ritvik Sahajpal,"Lecture Timetabling Using Hybrid Genetic Algorithms",IEEE , International Conference on Intelligent Sensing and Information Processing Proceedings, Vol.21. ,pp.24-34,(2004).

[6] Meysam Shahvali Kohshori and Mohammad Saniee Abadeh,"Hybrid Genetic Algorithms for University Course Timetabling",IJCSI International Journal of Computer Science

Issues, Vol. 9, Issue 2, No 2, pp. 446-455, March (2012).

[7] Thamilselvan, R. and P. Balasubramanie, "Integration of Genetic Algorithm with Tabu Search for Job Shop Scheduling with Unordered Sub sequence Exchange Crossover",Journal of Computer Science, Vol.8 (5):, pp. 681-693, (2012).

[8] R.Lakshmi, K.Vivekanandhan, R.Brintha, "A New Biological Operator in Genetic Algorithm for Class Scheduling Problem", International Journal of Computer Applications (0975 8887) Vol. 60 No.12, pp. 6-11, December (2012) .

[9] Amol C. Adamuthe, Rajankumar S. Bichkar,"Tabu Search for Solving Personnel Scheduling Problem", IEEE International Conference on Communication, Information Computing Technology (ICCICT), Oct. 19-20,(2012).

[10] S.N.Sivanandam, S.N.Deepa,"Introduction to Genetic Algorithms", Springer Berlin Heidelberg New York,2008

[11] Michel Gendreau, Jean-Yves Potvin, "TABU SEARCH.",2010

[12] Salwani Abdullah , Hamza Turabieh , Barry McCollum , Edmund K Burke, "An Investigation of a Genetic Algorithm and Sequential Local Search Approach for Curriculum-based Course Timetabling Problems", Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009) , Dublin, Ireland, pp. 728-731, 10-12, August (2009).

[13] Massoodian, Soolmaz, and Afsaneh Esteki. "A hybrid genetic algorithm for curriculum based course timetabling.", Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT'08, pp.1-11,(2008).

[14] Salwani Abdullah, Hamza Turabieh,"Generating University Course Timetable Using Genetic Algorithms and Local Search", IEEE, Third 2008 International Conference on Convergence and Hybrid Information Technology, Vol.01, pp 254-260, (2008).

[15] Cagdas Hakan Aladag and Gulsum Hocaoglu,"A Tabu search Algorithm To solve a Course Timetabling Problem", Hacettepe Journal of Mathematics and Statistics,Vol. 36 (1) ,pp. 53-64, (2007).

[16] Mushi, A. R., "Tabu search heuristics for university course timetabling problem", African Journal of Science and Technology (AJST)Science and Engineering Series Vol. 7, No. 1, pp. 34-40, June, (2006) .

[17] Sara K. Minery, Saleh Elmohamed, Hon W. Yaux,"Optimizing Timetabling Solutions Using Graph Coloring",NPAC REU Program Computer Science and Mathematics double major University of Dayton, pp. 99-106, (1995).

[18] Rushil Raghavjee, Nelishia Pillay, "Using Genetic Algorithms to Solve the South African School Timetabling Problem",IEEE, Second World Congress on Nature and Biologically Inspired Computing, in Kitakyushu, Fukuoka, Japan, pp.286-292, Dec. pp.15-17,(2010).

[19] Shengxiang Yang, and Sadaf Naseem Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling",IEEE Transaction on systems, Man, and Cybernetics-part C: applications and reviews, Vol.41, NO. 1, pp.93-106, Jan (2011) .

[20] Oluwasefunmi T. Arogundade, Adio T. Akinwale, Omotoyosi M. Aweda,"A Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem",International Journal of Computer Applications (0975 8887), Vol. 12 No.5,pp.1-4, December(2010).