AUTOMATIC TIME TABLE GENERATION BY USING PYTHON

[1]**A.Kalyan Kumar**, Assistant Professor, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[2]**G. Pratyusha**, Assistant Professor, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[3]**P.Divya**, UG Student, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[4]**Deepika,** UG Student, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[5]**M.Nithin,** UG Student, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[6]**Ch.Rakesh,** UG Student, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[7]**T. Nissi**, UG Student, Department of Computer Science and Engineering
DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India

ABSTRACT:
In this application, the automatically generated timetable reduces the complication of manually setting and managing a timetable. In implementation result we are utilize resource scheduling to decrease the difficulties of producing timetable. Our proposed method integrates a numeral of approach, intended to advance the cooperativeness of the explore operation. The time table does not overlap with their other schedules and these timetables efficiently utilized by faculty. In this work, we develop the application of time table which can be automatically generating time table according faculties' available time slots. This system provides benefits to the faculty and they do not need to worry for time clashes; a human do not need to perform permutation and combination and they can concentrate on other activities rather than wasting time by generating Time-Table. This system gives efficient time table generated according to professional college requirement.
KEYWORDS: Colleges, Time Table, Faculty, Courses, System, Constraints, Resource Scheduling, Optimal Solution.

1.INTRODUCTION:

Even though most college administration work has been computerized, timetable scheduling is still done manually due to the difficulties involved. The manual timetable scheduling requires considerable time and effort. Timetabling is the allocation, of given resources to objects that are placed in space time, in such a way that they satisfy a desirable set of objectives. The college lecture-timetabling problem asks us to find some slots and classrooms which satisfy the constraints imposed on offered courses, lecturers, classrooms and so on. The problem is a combinatorial optimization problem belonging to NP-hard class where the computational time grows exponentially as the number of variables increases. Various approaches have been made in the past decade to solve the problem of constructing timetables for schools and colleges. In our paper this problem is formulated as a constraint satisfaction problem and we discuss the various approaches that are capable of handling both hard and soft constraints. Hard constraints cannot be violated under any circumstances. For example, two classes cannot be allocated to a single teacher at the same time period, two classes cannot be attended by a student at the same time, more than one class cannot be held at a room at the same time et cetera. Soft constraints are necessary but not absolutely critical. For example, a timetable must be made in such a way that a group of students don't have to come to college to attend only one class.

2. Related Work:

There exist various timetable generation problems such as University Timetabling, Employee Timetabling, Sports Timetabling and Examination Timetabling. Carter and Laporte (1998) considered different categories to solve the timetabling problem. They are – Cluster method, Sequential method,

 DOI10.36893.JK.2023.V13I04N16.00203-00207

Meta-Heuristics and Constraint Based method. Meta Heuristics is a higher level procedure which is used to provide good enough solutions for optimization problems. On some class of problems, they do not guarantee a globally optimum solution. This method is used when the classical methods are too slow or fail to give a solution. This is achieved at the cost of optimality and precision for speed. In this paper we consider the following Meta-Heuristic methods. A. Genetic Algorithm (GA) Genetic Algorithms (GA) was invented by John Holland and has described this idea in his book "Adaptation in natural and artificial systems" in the year 1975 [6].Genetic Algorithms are inspired by Darwin's evolutionary theory. GA comes under the class of Evolutionary algorithms that use the principle of natural selection to derive a set of solutions towards the optimal solution. It is a search heuristic which generates solutions to optimization problems using techniques inspired by natural evolution like mutation, inheritance, crossover and selection. Here the algorithm is generally started with a set of candidate solutions called the population. Each solution in the initial population has a set of characteristics (its chromosomes or genotypes) which can be altered and mutated. Solutions from one population are taken and used to make another population, with a hope that the new population will be better than the old one [1]. Solutions are selected for breeding on the basis of their fitness. The fitness function usually identifies the number of constraints violated by a timetable. A timetable is said to be more fit if it violates less number of constraints. In the timetable generation problem, the population is a set of timetables maintained in memory. Each timetable is evaluated by finding the number of times it violates the constraints. Each timetable has an equal chance to participate in breeding. Algorithm 1: Genetic Algorithm GA ( ) [Start]Produce a random population of n chromosomes (suitable solutions for the problem) 1) [Fitness] Compute the fitness f(x) of each chromosome x in the initial population. 2) [New Population] Generate a new population by repeating the following steps until the new population is complete. (a) [Selection] Select two parent chromosomes from the existing population on the basis of their fitness. (Chromosomes with highest fitness are selected.) (b) [Crossover] Considering a crossover probability, crossover the parents to form new solution. (c) [Mutation] Considering a mutation probability, mutate new offspring at each locus (position in chromosome). (d) [Accepting] Add the new offspring in the new population. 3) Use the newly generated population for further run of the algorithm. 4) If the end condition is satisfied, return the existing best solution from the current population. 5) [Loop] Go to step 2. The efficiency of the genetic algorithm mainly depends on the fitness function [2, 8]. Mutation and crossover are the two main parts of the algorithm. They are called the operators of GA. The task of Crossover is the realization of the deterministic search [5]. From the previously selected pool, two parent solutions are selected for breeding. The new solution is obtained by the method of crossover and mutation and shares many characteristics of the parent solution. New parents are selected for every new child and this process continues until a population of appropriate size is generated. Only the best solutions from the previous pool are selected for breeding, along with a small number of less fit solutions to ensure genetic diversity. The creation of new population can be stopped when a solution which satisfies the minimum criteria can be found. In our case, the process can be stopped when a timetable satisfying all the hard constraints is found. Advantages: Provides diverse values of solutions and reaches global maxima. Mutation is used to induce diversity. Saving best solution is helpful. Disadvantages: It is complex to implement. B. Tabu Search Fred Glover proposed the Tabu Search in 1986. Tabu search can be directly applied to virtually any kind of optimization problem. Tabu search technique is one of the popular local search method based on neighbourhood search algorithm [2]. Tabu search basically avoids getting trapped at local maxima. That is why this search allows non-improving moves when it is trapped in local optima. Another advantage of Tabu Search method is that it prevents cycling back to the previously visited solutions by the use of memories thus creating more chances of improvement. Tabu search uses a search space which is the space of all possible solutions that can be considered. The search space could also simply be a set of feasible solutions to the problem. In Tabu search one of the basic elements is the "Tabu". Tabus are used to prevent cycling while moving away from local optima and through non-improving moves. Tabus are also used to move away from the previously visited.portions of the search space and thus help in exploring other regions. Tabus are stored in a short-term memory called the

 DOI10.36893.JK.2023.V13I04N16.00203-00207

Tabu list. Usually, the Tabu list is implemented as a circular list of a predefined length. The moves are considered as Tabu only for a certain number of iterations and this number is called the "tabu tenure". At each iteration the Tabu tenure value is randomly chosen in the range of 0.25Tb to 0.5 Tb (Tb is the square root of the total number of courses). Also, sometimes the best solution found so far using the Tabu search procedure may not actually be "the best". This is because Tabus are too powerful that they might prohibit attractive moves even when there is no danger of using those moves. This may lead to overall stagnation of the search process. Thus, a criterion called "Aspiration Criteria" allows cancelling the Tabus. The simplest and most commonly used aspiration criterion is allowing a move, even if it is Tabu [1]. The overall quality of the timetable is evaluated based on a function of soft constraints' violation called the objective function. The objective function f (q) is defined as: f (q) = ∑ wi qi (1) where wi and qi are the weight and violation number of the ith soft constraint, and i ranges from 1 to n (n is the total number of soft constraints). A neighbourhood solution satisfies aspiration criterion if and only iff (Qnei) < f (Qbest) (Qbest is the best solution found so far). Algorithm 2 : Tabu search - F(x) 1) Building an initial solution using greedy algorithm. (a) Non-preassigned courses are split into blocks and are assigned to appropriate courses. (b) Non-preassigned courses are assigned to appropriate rooms based on results of (1). 2) Applying Tabu search algorithm to improve the initial solution. Some of the courses may be preassigned, that is their periods and rooms are predetermined by people, while the other courses are not. These preassigned courses are not re-scheduled again. The searching process is stopped when it meets one of the following criteria: The number of iterations passed is beyond a certain predefined number or when the best solution found so far satisfies all the constraints or when the number of consecutive unimproved iterations has reached a certain number [6]. The results of the timetable obtained from the Tabu Search algorithm shows significant improvements when compared to handmade results. Therefore, automated timetables obtained from Tabu Search algorithm are much better than handmade timetables due to the high improvement rate. Advantages: It is faster than any genetic algorithm and reaches local maxima. Also it avoids local optima. C. Iterative Local Search The optimization problems and combinatorial search are tackled by Local Search meta-heuristics are an emerging class of methods, which was recently proved to be very effective for a large number of combinatorial problems. The Local Search techniques are based on the iterative exploration of a solution space. At each repetition, the algorithm for Local Search moves from one solution to one of its "neighbours", i.e., solutions that are (in some sense) close to the starting one. One major drawback of this family of techniques is the lack of robustness on a wide variety of problems. In fact, in so many cases, these kinds of methods guarantee finding good results in practicable run times, while in other cases Local Search techniques are caught in the local minima. Combination of several neighbourhood structures is one of the alternative approaches [2, 3] to cope with local minima. A set of neighbourhood operators are given such that for given collection of basic neighbourhoods, a new compound neighbourhood is automatically created and prescribe the strategies for its exploration. This approach is based Multi - Neighbourhood Search.

2.1Scope:

- Colleges and Universities can employ these to generate timetable automatically.
- Productivity and manual lab can be decreased due to increase in working speed by using an automated system.
- It can be used for scheduling in schools and even in other organizations to schedule their employees based on resource management.

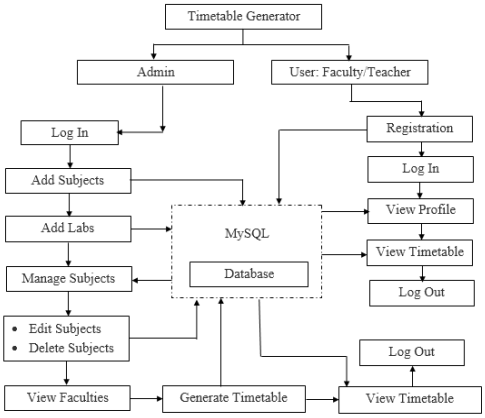
3.Proposed method:

We propose this system that can be consiimprove hemimetaboly[10] providing support through the Timetable Management System, timetable can be generated for any faculty without any overlap. The system is developed in a Flask based Python environment. MySQL is used for database management.

3.1 Flow of the project:

 DOI10.36893.JK.2023.V13I04N16.00203-00207

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023



4.RESULTS AND DISCUSSION:

Our training collection now includes approximately 400 different frames examples that were pulled from the internal and added to it. To save time and effort on the computations, we keep track of the changes in intensity across the global at a rate of one change recorded every two seconds. The utilization of the RGB channels allows for the recording of colour tones to be successfully accomplished. If there is a discernible shift in the overall luminance of an image, the tone mapping function will need to be recalculated for each colour channel before it can be saved to a look up.

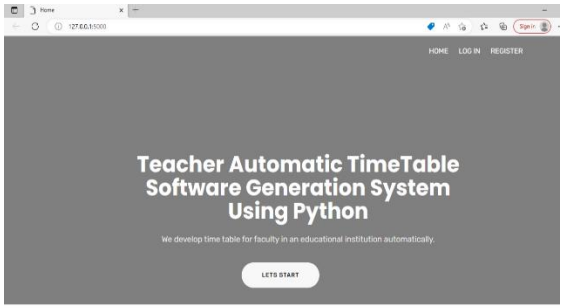


FIG:1 HOME PAGE

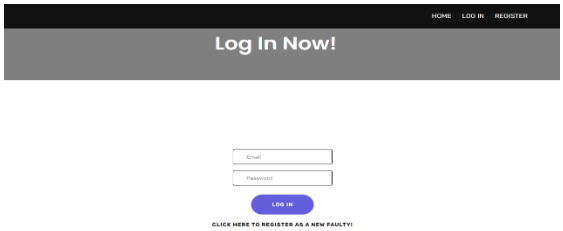


FIG:2 LOGIN PAGE

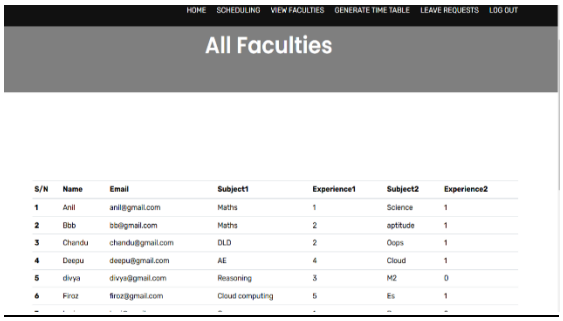


FIG:3 ALL FACULTIES

 DOI10.36893.JK.2023.V13I04N16.00203-00207

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023

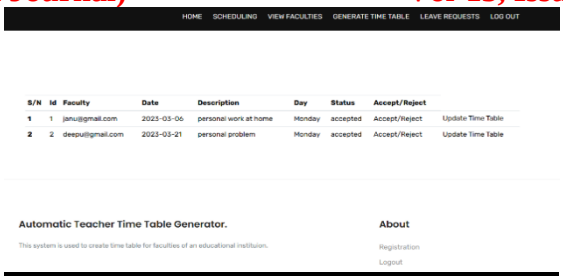


FIG:4 ADMIN LOG IN PAGE

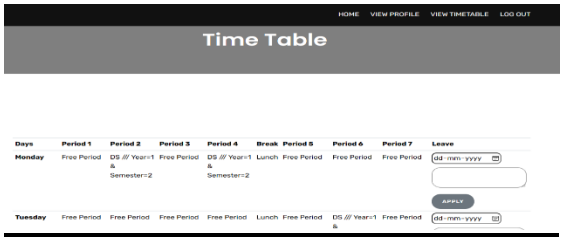


FIG :4 TIME TABLE

5.CONCLUSION:

In this application, we have successfully created a system to generate time tables for teachers. This is developed in a user-friendly environment using Flask via Python programming. The admin is like a dean of a college where he/she can assign subjects and labs to each class. The admin is also able to generate timetable from scratch and store it in the database. The user, on other hand represents a faculty or a teacher who needs to register can is allowed to view his/her own profile and weekly timetable. The faculty can only view their own time table whereas the admin can view details about all the users.This system minimizes the time and effort needed to develop a timetable.

REFERENCES:

[1] A, Parkavi. (2018). A STUDYONAUTOMATICTIMETABLEGENERATOR. International Journal of Innovative Research & Growth. May.
[2] Srinivasan, D. & Seow, Tian & Xu, Jian-Xin. (2002). Automated time table generation using multiple context reasoning for university modules. Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002. 2. 1751 - 1756. 10.1109/CEC.2002.1004507.
[3] M, S., Vaze, P. K., Pradeep, & N R, M. (2017). Automatic Time Table Generator. International Journal of Advanced Research in Computer Science and Software Engineering, 7(5), 204–211. https://doi.org/10.23956/ijarcsse/sv7i5/023 [4] Abramson, D. (1991). Constructing school timetables using simulated annealing: sequential and parallel algorithms. Management Science, 37, 98-113.
[5] Abramson, D. (1991). Constructing school timetables using simulated annealing: sequential and parallel algorithms. Management Science, 37, 98-113.
[6] De Werra D., “An introduction to timetabling”, European Journal of Operations Research”, Vol. 19, 1985, pp. 151-162. De Werra D., “An introduction to timetabling”, European Journal of Operations Research”, Vol. 19, 1985, pp. 151-162.
[7] Carter M. W., Laporte G., “Recent developments in practical course timetabling”, Lecture Notes in Computer Science, Vol. LNCS1408, Springer-Verlag, 1998, pp. 3-19.
[8] Schaerf, A., “A survey of automated timetabling”, Artificial Intelligence Review, No. 13, 1999, pp. 87-127.
[9] Nandhini, And S. Kanmani, “Implementation of Class Timetabling Using Multi Agents”, (2009).
[10]. G.Sai Chaitanya Kumar, Dr.Reddi Kiran Kumar, Dr.G.Apparao Naidu, “Noise Removal in Microarray Images using Variational Mode Decomposition Technique ” Telecommunication computing Electronics and Control ISSN 1693-6930 Volume 15, Number 4 (2017), pp. 1750-1756

 DOI10.36893.JK.2023.V13I04N16.00203-00207