JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Automatic Timetable Generator Using Genetic Algorithm

Parikshit Shinde
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India

Pratik Landghule
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India

Archana G. Said
Assistant Professor
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India

Pranav Saykar
Department Of Computer Engineering
AISSMS Institute of Information Technology

Tanmay Janrao
Department of Computer Engineering
AISSMS Institute of Information Technology

Abstract - **While creating a schedule can be challenging, it is crucial in educational institutions. At the moment, the timetable is manually handled. All the periods will be managed automatically, and it will be useful for faculty to have a timetable on their phone via an application. When a teacher is missing, tardy, or early, the schedule will still be managed. For the purpose of efficiently creating a schedule, the maximum workload for a faculty for a day, a week, and a month will be set. It takes a lot of effort and time to make a schedule. In our paper, we attempted to lessen the problems associated with establishing a timetable using a genetic algorithm. In addition to logging into the system, the system administrator also inputs the courses with their codes and the unit The next runner is where all the lecture halls or flats that will be used are entered after the courses have been entered. The system also generates the schedule system after they are entered. This method (inherited algorithm) aids in minimising crimes and errors that may be encountered when creating an automatic timetable. We are able to shorten the time needed to create a timetable and create a schedule that is more exact, precise, and free of fatal offences by employing inheritable algorithms. The institute's common mandatory classes are all included in the first phase and are listed by a central platoon. Individual departmental classes are contained in the alternative phase. Currently, the only way to manually set this schedule is through altering earlier ones.** I.

INTRODUCTION

A timetable is a type of schedule that specifies the times that particular events are supposed to take place. The primary task when an educationalist tries to manually create a schedule is to confirm the work and availability of the faculty members for a particular subject. The creation of a schedule for each course for a given semester while keeping in mind the established university schedule is the other major task. Each subject's time slots should be evenly distributed across a semester such that no faculty member's time slots from one semester overlap with those from another. The project's proposed approach will make it easier to manage the Schedule without having to repeatedly set it up manually. The system will use machine learning and inputs such semester-by-semester subjects, faculty, and workload of the faculty member.

It will produce a potential timetable for the working days of the week based on these inputs. The proposal is made in "Time Table Scheduling using Genetic Artificial Immune Network." One of the crucial duties that arise in everyday life is scheduling. Many scheduling issues exist, including those involving employees, production, education, and others. Due to the many requirements that must be met in order to find a workable solution, organising educational schedules may also be challenging. Techniques like genetic algorithms have been employed with varying degrees of effectiveness. Scheduling, often known as timetabling, is the practise of allotting time for organised planned actions to produce an outcome that is satisfactory and unrestricted. Transportation, sports, workforce, course, and exam scheduling are some of the use case scenarios.

LITERATURE SURVEY

J.J Grefenstette [1] evolutionary approaches to the time scheduling problem. styles comparable to the evolutionary algorithm set and genetic algorithms. mixed results while in usage. In this essay, we've examined the challenge of using a genetic algorithm to catalogue an instructional timeline. A synthetic inheritable defensive network and a non-original mongrel algorithm were also used to solve the issue, and the outcomes were compared to those obtained using the inheritable algorithm. The findings demonstrate that GAIN is more suitable than GA for arriving at a potential conclusion quickly.

Anuja Chowdha [2], Academics continually struggle with the issue of discovering the study schedule that is feasible at the university's major department. This research provides an evolutionary algorithm (EA) technique built on solving the resilient schedule problem at the university. On to chromosomal representation issues. At the correct computer moment, heuristics and contextually grounded thinking may have been attained. Cohesion has been improved by the clever usage of inheritable revision. Using actual data from a top university, the full class plan described in this paper is accepted, estimated, and discussed. An automatic timeline system that.

Anuja Chowdhary [3] developed uses an efficient timing algorithm that can handle both strong and weak barriers. After a specific semester is over, each teacher and student can look at their timetable, but they shouldn't plan ahead. According to the teacher's schedule, the availability and power of visual resources, and other rules relevant to various courses, semesters, preceptors, and grade positions, the Timetable Generator System develops a timeline for each class and teacher.

Anirudha Nanda [4] proposes a typical resolution to the timing issue. most difficult past heuristic programmes that have been suggested from a scholarly perspective. This outcome is still valid when viewed from the perspective of the topic, specifically the teacher's inadequacy at a certain moment. The planning result described in this work is adaptable, with the main goal of resolving academic and academic dispute, schoolteacher-related concerns. All implicit walls (e.g., schoolteacher vacuity, etc.) are replied firmly.

Al-Khair [5] suggested using algorithmic techniques to solve the timing issue with teacher availability admissions. The challenge of scheduling academy time is completely addressed by this technique, which employs a heuristic approach. First, it creates a temporary timeline using randomly generated title sequences. If a teacher divides a class into more subjects than are allowed, the extra subjects are moved to the Conflict data structure. By taking into account the tight relationship between the schedule problem and the vertex colouring problem,

Csima and Gotlieb [6] also addressed the schedule problem as a three-dimensional assignment problem. The now wellknown particular link between the diverse scheduling difficulties was originally examined in this study. Becker deceived their respective executions with "hand" computations, extending the work These publications typically relied on a heuristic methodology. This work led to a large number of papers that examined the issue but didn't add any fresh insight..

Broder [7] asserted that the objective function can be minimised by repeatedly assessing a collection of relevant nonlinear equations (deduced from arbitrary or montecarlo assignments). It was asserted that the outcomes thus obtained might not be ideal but rather would be comparable to the initial minimal outcomes obtained by prior heuristic methods.

Chan H [8] as being crucial for the functionality of Gotlieb's set partitioning system of schedule generation. However, the first algorithm itself experienced duplicated effectiveness as a result of this. Lions added a conception system for the creation of class/teacher calendars to this basic task. In his paper, he discussed how to apply Kuhn's Hungarian system to the matrix reduction issue needed by Gotlieb's schedule generation method.

Walker and Macon [9] They implemented a Monte Carlo algorithm that randomly chooses classes for students, illustrating the difficulty of placing students in a set of courses. They observed that the application of the randomising procedure had a limit on the ultimate best outcome.

Punter [10] His method entailed transferring this data onto rendering wastes before it was processed by a computer, as well as manually producing allocation plans that outlined the instructional circumstances for each class. The discovery of contradictory sub-sets of assignment conditions was made possible by analysis employing a graph colouring approach. It was claimed that the system might be used to generate workable calendars with the appropriate emendations

Hulskamp [11], who created software to assist engineering students in investigating a wide range of potential solutions to a design problem. The software was put into use at Caulfield Institute of Technology, and it was successful in illuminating the problem's complexity and phase space. This may be the first publication of a DSS for the creation of schedule results.

Deb K [12] created a programme that was not based on integer programming but was also effective and effective. As a DSS, it made it simple for users and the programme to collaborate on finding the best solution. Calendars were successfully constructed (as opposed to generated) according to empirical tests.

Wright M [13] discussed the use of mathematical models to achieve organizational goals. They mentioned two specific approaches: mixed integer programming and thing programming. Mixed integer programming involves using variables that can only have whole number values in the solution. Thing programming, on the other hand, involves using optional variables called slack variables that can help improve the optimal result, but are not required

Hemmerling [14] developed a program to create schedules for South Australian seminaries. This program conducted a comprehensive search to determine the success of finding a solution to a specific timetable problem in the academy. The study reported several findings and highlighted the significant amount of computer processing time required. The tests were performed using the CDC 6400 computer, often utilizing double word patterns. The program was quickly adopted by many secondary schools in the state at that time. However, due to its reliance on a complete treespanning approach, it is expected to be outperformed by the combinatorial advancements in the phase space

Knauer [15] introduced a program that assessed outcomes and used an exchange process to improve them within specific limits. The algorithm had a positive impact on the quality of school timetables, leading to significant improvements. Major high schools in Wurzburg implemented these computer-generated schedules. However, the discussion did not address the challenging nature of the broader problem or its intractability.

ALGORITHM WORKING

Genetic Algorithm-The inheritable algorithm, which is based on natural selection, the process that drives natural elaboration, is a method for solving both limited and unconstrained optimization problems. A population of individual results is continuously modified by the inheritable algorithm.

Algorithm Working-

Steps Involved- 1.Initial Population-

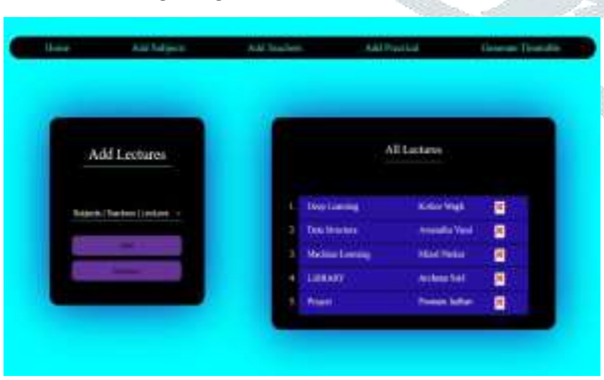
Initially, we were supposed to create the first-year class schedule by entering all of the section-by-section information. Read the database's faculty, subjects, and section information. Here algorithm will generate first possible solution by randomly assessing inputs known as chromosomes.

2.Fitness Function-

In this step we will consider all the inputs such as teachers classrooms timeslots and students and assign them fitness value as and try to arrange them such as there is no clash between various inputs like there might be some extra classrooms which are available thoroughly during college session we can accommodate those classes to any particular timeslot. Fitness function plays very crucial role as it is a building block on which our algorithm is going to rely on.

3.Selection-

Selection step selects the best possible solution containing students teachers subjects classrooms and timeslots and the Chromosomes within the population are chosen by this driver for reduplication. It is more likely to be chosen for breeding over time the better the chromosomal fit. A fraction of the accessible population is chosen to elect a new generation for each succeeding product. A fitnessbased approach is used to select each individual result, with fitter results often having a higher likelihood of selection.

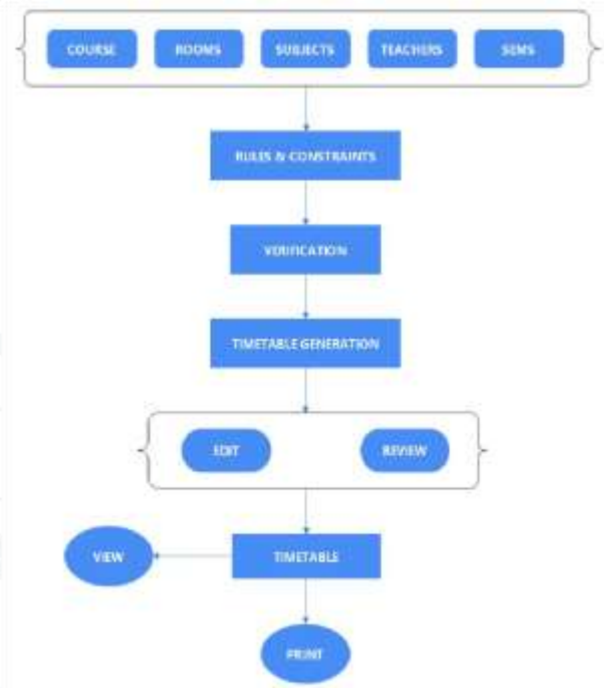


4.Crossover-

Crossover takes the best possible solutions produced by the step 3 but there might be still some clashes between some constraints like time slot overlapping or assigning same classroom for two different subjects at same time also sometimes same teacher could be assigned to different classes at same time we solve this problem in this step.

5.Mutation-

Mutations introduce randomness in the results and it maintains diversity ultimately resulting in avoidance of premature termination like sometimes teacher might not want to teach in particular class also some professors might prefer larger classrooms inspite of the smaller number of students mutation takes cares of such situations results might not always be perfect.



ARCHITECTURE OF THE SYSTEM

The system's capacity to input the different courses, lecture halls, departments, programmes, lecturers, and the definition of a few limits from which the timetable is built reduces the high cost and lengthy turnaround necessary in the creation of close to ideal timetables.

Timetabling genetic algorithm approach scheduling a Timetable for college –

As mentioned, scheduling classes for a council schedule is constantly encountered With constraints(Hard and Soft) due to diversity as compared to a academe schedule where conditions are largely limited. The problems associated with the hard constraints need to be resolved to produce a functional result. To optimize the performance of the of the scheduling it is important to address the issues linked to soft constraints. Still, a careful approach should be formulated without compromising the affect to hard constraints to minimize serious disruptions to the system. As analogous, a straightforward scheduling system as for without switching to a different strategy, a tiny academic system cannot be applied to a complicated association in a way that produces quick and effective outcome. Several soft restrictions may clash and as a result, a compromise will need to be made between them. The soft restriction will assign an appropriate classroom that can fit the students. For instance, a class might have 12 students. Yet, the lecturer might prefer a room with capacity of the 45 students. As a professor choice is taken into account in the soft constraints, it is hoped that the class scheduler will identify a proffered configuration

The following information will serve as the foundation for the class scheduling issue.-

available academics accessible cells Schedules for student groups There may be free time intervals on a council student's timetable, which sets it apart from a grade- academe schedule. This is based on how many classes the scholars take. The class scheduler will designate a time slot, a teacher, a location, and a student group for each class. By multiplying the number of student groups by the number of modules each student group is registered in, it is possible to calculate the total number of classes that need to be planned. The following hard limitations will be taken into account for every class schedule produced by this process.

Only free classrooms allow for the listing of classes. A professor is only able to instruct one class at a time. The student group must fit in the classroom, thus it must be large enough. A particular number of class packets are required while coding the class schedule. The class's scheduled time, the instructor's instruction of the group, and the necessary classroom are the classes

time space niche lecture course or module RoomID, Room Number, Room Capacity, Course/ ModuleID, Course/ Module Code (Sections), and Course/ Module Timeslot ID and Timeslot Professor Name and Professor ID. ID, Professor. Group ModuleID, GroupID, and Group Size CClass Denotes the student's current class. IDs for the class, group, module, professor, timeslot, and room. All of these items will be recapped in the timetable. This class will be useful for separating the interactions between various restrictions. Additionally, it will be in charge of parsing chromosome for use as an example in an instruction algorithm to create a candidate timetable that can be estimated and graded.

CONCLUSION

Handling faculty members and physically allocating lectures to them is a difficult task. So, our suggested system will aid in overcoming this drawback. As a result, we are able to create a schedule for any number of courses over several semesters. This system is simple to use and offers quick and efficient schedule generation, saving time and resources. The layout lessens the time commitment and discomfort of manually creating the timetable.

REFERENCES

[1] J.J Grefenstette, editor. Proceedings of the First International Conference on Genetic Algorithms and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of The 6th International Conference on the practice and Theory of Automa.

[2] AnujaChowdha ry, PriyankaKakde, ShrutiDhoke, SonaliIngle,RupalRushiya, Di- nesh Gawande, ’Time table Generation System”, International Journal of Comp.

[3] Anuja Chowdhary A. White Paper: ”A hybrid GA-heuristic search strategy.” AI Expert,USA.

[4] Anirudha Nanda, Marco Dorigo, Vittorio Manniezzo (1992). ”A Genetic Algo- rithm to Solve the Timetable Problem” Journal of Computational Optimization and Applications, 1, 90-92.

[5] Al-Khair ,Fischer T., Gubbels H., Hacker C., Hasprich O., Scheibel C., We- icker K., Weicker N., Wenig M., Wolfangel C. Automated solution of a highly constrained school timetabling problem - preliminary results. EvoWorkshops, Como-Italy.

[6] Csima and Gotlieb ,Elliman D and Weare R .”A genetic algorithm for university timetabling system.” Presented at the East-West Conference on Computer Technologies in Education, Crimea, Ukraine.

[7] Broder ,A multiobjective genetic algorithm for the class/teacher timetabling problem.” In Proceedings of the Practice and Theory of Automated Timetabling (PATAT’00), Lecture Notes in Computer Science,Springer, 2079, 3-17.

[8] Chan H. W.”School Timetabling Using GeneticSearch.” 2th International Con- ference on thePractice and Theory of Automated Timetabling, PATAT’97.

[9]Walker and Macon”An updated survey of GA- based multiobjective optimization tech-niques.” ACMComputing Surveys, 32(2), 109-143.

[10]Punter”A tabu search algorithm for computing an operational timetable.” European Journal of Operational Research, 76(1), 98-110.

[11] Hulskamp., Deb K., Fonseca, C.M. (2006). Multi-objective evolutionary algorithm for university class timetabling problem, In Evolutionary Scheduling, Springer- Verlag Press.

[12] Deb K., Agarwal S., Pratap A., Meyarivan T. (2002). ”A fast and elitist multi- objectivegenetic algorithm: NSGA-II.” IEEE Transactions on Evolutionary Computation,6(2), 182-197.

[13] Wright M (1996). ”School timetabling using heuristic search.” Journal of Oper- ationalResearch Society. [14] Hemmerling ”Ant Algorithms for the Exam Timetabling Problem.” 6th International Conference on the Practice and Theory of Automated Timetabling, PATAT’06..

[15] Knauer . ”The construction of class-teacher timetables.” Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam.