Grenze International Journal of Engineering and Technology, June Issue

Smart Time Table Generation using Artificial Intelligence

Sukhwant Kour Siledar[1] and Dr. Vijaya B. Musande[2]
[1]M. Tech, Department of Computer Science & Engineering, JNEC, MGM University, Aurangabad
[2]Professor, Department of Computer Science & Engineering, JNEC, MGM University, Aurangabad
sukhwantkour211998@gmail.com, vijayamusande@gmail.com

***Abstract*— For any educational institutions be it a small-scale school or a college or huge institutions like universities, time tabling is a very important and tedious task. It concerns all activities with regard to producing a schedule that must be subjective to different constraints. Being dependent on various constraints, it is a temporal arrangement of courses, faculty and students hence it requires changes every now and then. Creating such time tables manually especially with large scale institutions like universities, leaves scope for clashes and human errors. To overcome these problems automated time table generation using AI can save a lot of time of administrators and manpower. Such an automated system will require various inputs like course details, infrastructure, teachers available and strength of class, and will aim to make most optimized utilization of all these resources in a way to best suit any of constraints of college rules. A precise time table will then be chosen from generated solutions. Time table is scheduled for various purposes like organizing lectures, bus schedules, exam time tables and many more. In our proposed model we will make an approach to build a model that can automate time tables to serve various scheduling purposes using Evolutionary Algorithm (EA) and Genetic Algorithm (GA), and further can be integrated with ERP of institutes for ease of data fetching. This technique allows one to make an attempt to create time table with reduced errors and mistakes in the least time.**

***Index Terms*— Timetable, Generation, Genetic, Optimization, Constraints, Scheduling, Fitness, Chromosome.**

I. INTRODUCTION

Timetable is basically a structure which shows the time at which some prescribed event occurs. For educational institutes timetable is for achieving its basic purpose of lecture delivery, and is used for scheduling of events throughout the day, week, term or year for each batch. It requires the combination of resources like batch of students, classes, instructors, time slots, and days arranged in a way such that no mentioned resources have an overlap. This practice of mapping events in general (classes/ exams) to timeslot subject to the constraints, is carried out manually in most of the institutes, requiring lot of manpower and time. Hence, time tabling gives rise to scheduling problem that is tedious and as well requires solution in every institute at least once or twice an academic year.

From above discussion, time tabling is a non-polynomial (NP) complete problem, i.e., a problem which has no defined way to draw an appropriate solution. This NP complete scheduling problem falls in the class of computational problems for which no efficient algorithm that can give accurate solution has yet been found.

Grenze ID: 01.GIJET.9.2.336

Hence to provide an efficient solution to the stated problem, we will make an adaptive heuristic approach, which will generate a set of good solutions from which the most optimized solution will be provided as output. In the field of computer science, for precision or optimization problem, artificial intelligence is used to implement a model providing solution to it.

The main objective of this paper is to address the timetabling problem (which is NP complete, scheduling problem) by using AI which will result in an optimal solution with minimal or no redundancy errors. This model is developed using genetic algorithms falling in the class of evolutionary algorithm of AI. It works in adaptive heuristic searching way for solving constrained optimization problem. This approach is based on natural selection and then proceeding for evolution of selected individuals from population to reproduce generations i.e. pool of many timetables here, from all these generations on the basis of fitness the most optimal solution is given as output. The classical GA that we will be using in our model is based on Darwin's theory of evolution and the principle of survival of the fittest.

II. Related work

The constant struggles in designing of time table have been observed widely associated with small to large scale educational institutions. While this area is vast covering scheduling of regular classes, examinations, bus schedules and many more; various efforts have been made globally to address this problem and lot of research has been carried out to propose and implement various methods for automated generation of time tables. As also mentioned in the previous section time tabling is a NP complete problem, so only the attempts to the best possible solutions have been made so far. Optimal solutions were achieved using both kind the traditional and artificially intelligent techniques. This chapter will examine all the variety of approaches carried out so far.

The stated timetabling problem was first studied by Gottlieb in 1963, who considered that each lecture consisted of one student group, and one teacher whose combination can be chosen freely. Since then, the problem is being addressed by proposing various models using variety of methodologies to develop a computational solution which can replace manual task of scheduling, as discussed in the following part of this section.

Authors Dipti Srinivasan Tian Hou Seow Jian Xin Xu (2002) [1] proposed a model for automated timetable generation using multiple context reasoning. They presented an EA based approach along with context-based reasoning for creating physical timetables in less computational time. But it is difficult to implement as compared to GA and has less accuracy and optimization.

Shengxiang Yang, Member, IEEE, and Sadaf Naseem Jat (2008) [2] defined university course timetabling problem (UCTP) as a combination of events and time slots. They addressed it using GA with Local Search (LS) technique. The GA guided search strategy create offspring based on individuals from previous generations and LS use its exploitive search to improve the efficiency and quality.

D. Nguyen, K. Nguyen, K. Trieu, and N. Tran (2010) [3], used Tabu search algorithm to develop the model for giving solution to timetabling problem. In this, the search space comprises of set of achievable solutions. A fundamental "taboo" element exists to move aside the non-improving moves and gradually it keep away from getting trapped at local maxima but this is expensive approach for evaluating the resources and formulating it.

N. M. Hussin and A. Azlan (2013) [4], implemented graph colouring heuristic method for scheduling problem. Here problem stated is represented using graphs for constructing stages of difficulty in the process of scheduling. Nodes of graph denote subjects and edges denote conflict. With each phase it keeps improvising the solutions and ultimately reach the best solution, but the prolonged time taken to reach this stage makes this method less reliable.

Authors W. F. Mahmudy and R. E. Febrita (2017) [5] used fuzzy logic method for providing computational solution to time tabling problem. This multivalent logic is based on fuzzy set theory and linguistic variables are used to solve optimization problem of timetable and provide a realistic solution. But this makes it difficult to evaluate membership function and ultimately it is hard to create and calibrate fuzzy model.

T. Elsaka (2017) [6], used Constraint satisfaction modelling which is based on constraints and variables and not on the object function. Two main components here are constraint and data. Constraint programming has statement of constraints that serve as part of program which is an advantage of this model. The main hindrance is the amount of time it consumes and the soft constraints are not considered.

Authors K. Y. Junn, J. H. Obit, and R. Alfred (2018) [7] proposed a solution based on GA which is based on theory of natural selection. It is an iterating process of creating population from individuals. Convergence can be a drawback if not taken care in the iteration.

D. Apostolou and E. Psarra (2019) [8] have proposed an approach on basis of Hybrid Particle Swarm optimization (PSO) with local search, which is an AI method. The stated problem is provided a solution by

integrating PSO with prototype methodology which creates particles that can upgrade themselves and have own memory. But unlike GA it does not have operators like crossover and mutation to avoid convergence.
This section explained how others through various approaches used intelligence to solve the problem by setting rules and how classical genetic algorithm can prioritize these rules dynamically to optimize the timetable generation by providing benefit of distributed solution and load balancing. It can serve as the best possible way to provide a solution provided constraints and proper convergence condition.
From above discussion, many studies have focused on making an approach to time table scheduling using AI by considering diverse techniques. However, rapidly evolving demands do not sustain with these attempts. The well-established relationship between constraints and scheduling is very crucial for our problem to provide an optimized solution. In light of this, we also studied the area of genetic algorithms in great depth, researchers have made latest and advanced studies that can be compatible with the evolving world.
Indeed, a brief analysis by Sourabh Katoch, Sumit Singh Chauhan, Vijay Kumar (2020) [9] on advances in genetic algorithm and their implementation, has made a clear differentiation between all above discussed attempts and motivated us for an approach using GA in our problem domain. Further for GA implementation and mathematical modelling, the analysis made by L V Stepanov, A S Koltsov, A V Parinov, A S Dubrovin (2018) [10] was studied and on its basis our proposed model was developed.

III. METHODOLOGY

A. Proposed System

In our approach we have developed model for providing solution to Timetabling problem based on genetic algorithm in AI, it is based on natural selection and evolution method. Here we use basic terminology which can be referred as follows.
Phenotype refers to population in real world, whereas Genotype is used for reference to population in computational world.
Population is generally referred to set of human beings in phenotype but in genotype it is set of solutions (here it will be a set of generated time table, also sometimes population can be referred to as generation)
Chromosome is the individual solution to given problem and a gene specifies one element position in a chromosome.
To simply understand implementation of GA and the genetic operators that are responsible for the alteration in composition of offspring, the following part of this section can be referred.
Implementation of GA in our developed model is done as in below steps,

Preprocessing
The prerequisite to perform operations in GA is to convert potential solution into a simple value like a string of real or binary numbers. It helps in improvement of speed of algorithm. We have used conversion of data to binary string i.e., a string of combination of 0 or 1. These bits of string are responsible for characteristic presentation of solution as well as algorithm accuracy. Each chromosome string is a composition of sequential arrangement of gene string.

Initial population
After encoding, the first step of the algorithm is to generate initial population which is done by random creation of individuals on the basis of the constraints defined. The larger the population, the better will be the results. This process sometimes can also be described as selection process or in terms of GA it can be referred as reproduction operator.

Evaluation of population
The parameter used for evaluating an individual is known as fitness of an individual. The fitness of each chromosome is determined within a generation, which is an estimate of how well the solution satisfies the given constraints relative to other solutions.
As we have used binary encoding, our range for the fitness function will be between 0 (worst solution) and 1 (best solution). The fitness function defined in our approach is as follows,

$$F(a) = \frac{\sum_{i=1}^{n} a_i * h * d}{t}$$

where, a= timetable solution under evaluation,

h= hours per day,

d= days per week,
t= total fitness of a generation

After determining the fitness of all chromosomes in a generation, i.e., complete evaluation of population, we need to select chromosomes for further mating so that new generations can be created. This is done using Roulette wheel selection method [9]. The basic principle on which this selection method works is

Selection α fitness

The concept of this technique, is to dividing a wheel according to proportion based on fitness value. Then each chromosome is mapped in the suitable proportion. Eventually, the wheel is rotated and the pointer points at the chromosome is selected for further reproduction. Also, as the basis of proportion is fitness value, mostly the larger proportion will be composed of fitter chromosomes, increasing probability of pointer to stop at fitter individuals.

Crossover

Now for mating the selected chromosomes, the crossover operator is used, hence resulting in new chromosomes together making a new generation. We have used single point crossover in our model. The selected chromosomes using above selection method are used and single point crossover is performed [9]. In this operation, random point of a chromosome is selected, and for two parents, gene after this point is swapped resulting in new off springs with different gene composition.

Mutation

This genetic operator is very crucial as it is used to prevent converging at a very early stage of reproduction. It basically alters genes of chromosomes for generating diverse variety of population. We have implemented Displacement mutation (DM), as name suggests it operates by displacing genes within a chromosome. This results in production of diversity, hence reducing risk of premature convergence.

B. Algorithm

Based on the above described implementation of GA in our model, its algorithm can be depicted as in Fig. 1.

C. Constraints

In GA constraints can be classified into two categories, *Hard constraints*, these mandatorily needs to be followed. In our model the hard constraints that are,

- Same student must not have two lectures simultaneously.
- Same lecturer must not have two lectures simultaneously.
- One room must be allocated to only one lecture at a time.

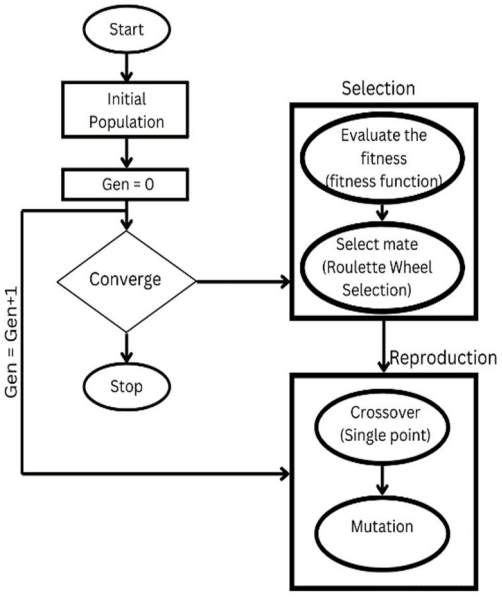

Figure 1. Algorithm

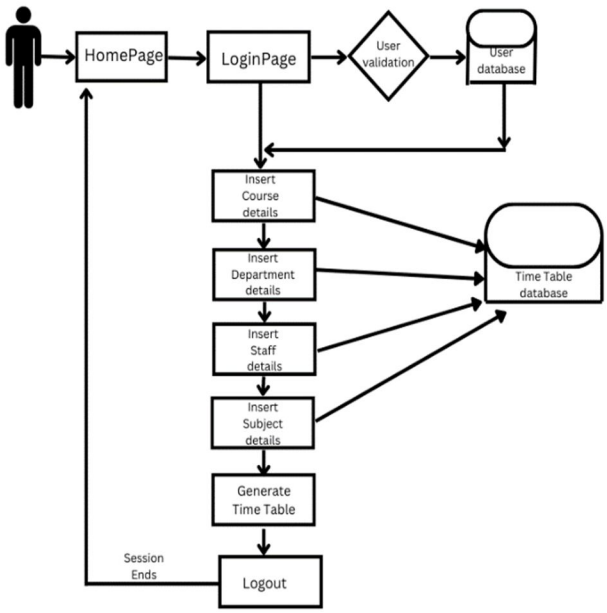

Figure 2. System architecture

Soft constrains are the constraints that are required to be satisfied, but it is not obvious that they will always be satisfied. In our model the soft constraints considered are,

- Same lecturer must not have two consecutive lectures.
- Fixed slot assignment for particular subject.

D. System Architecture

The system architecture of the developed model as shown in Fig. 2 uses GA to solve timetabling problem and generates optimal minimum or no error solution. It has capability to take various resources (class, subject, teacher details) as input in a very user-friendly manner and process them for the required output in low cost and less time.

E. Hardware and Software Requirements

The minimum *hardware requirements* for implementing the system are Processor of minimum configuration Pentium IV/Intel I3 core with speed of at least 1.1 GHz, RAM 512 MB (min) and Hard Disk of minimum 20GB. The output is displayed on standard monitor screen and for data input keyboard and mouse is required.

Software requirements for implementation is about the prerequisites to be installed on the system for proper functioning. In our model, such prerequisite software are Java 8 and above supporting compiler, Struts-2 framework, Apache tomcat server and MySQL database. Operating system supporting our model are Windows 7 and above versions.
For developing front end Html5, CSS, JavaScript, bootstrap and AJAX is used, whereas the backend code is in Java.

F. Modules and Interfaces

- *Registration and Login:* When the user visits the homepage, for the first time they need to register by providing basic details. These are then stored in the user detail database as seen in system architecture Fig. 2. Login is a submodule of this interface, where user can enter credentials for logging in.
- *Input Interface:* After login, user is asked for all input details required like number of slots, batches, days per week for which schedule is to be created. Then after the course details along with faculty details and submit button is clicked. This directs to the browser page where the optimized timetable is displayed.

IV. RESULT ANALYSIS

Timetable is generated using genetic algorithm and this optimized solution satisfies all hard constraints defined and most of the soft constraints.
In Fig. 3, the snapshot of the console shows the fitness values reached in various generations by the chromosomes. All of the generations are presented on the console itself in the genotype and on the output screen, only the phenotype optimized solution is displayed. Also, by clicking on refresh we are able to get the lesser optimized solution on display, this is the facility for convenience of human to compare and contrast given solution with the other possible solutions if in any case human intervention is needed to alter the produced timetable. Results displayed are free from clashes and generated in a very less time as compared to manual creation and also saves lot of efforts.

V. CONCLUSION

We have implemented GA in AI for smart time table generation, which produces optimal timetable subject to the constraints defined.
Though the solution cannot always be 100% optimal and suited, as the degrees of optimization depend upon the constraints defined. The improvement has been achieved using intelligence system by the appropriate use of genetic operators. Developed system works accurately initially for schools as they execute classes in a very classical model and gradually increases its scope to colleges and higher secondary classes which has more constrains hence making the problem more complex, but our model deals with it in a user-friendly way. There is future scope to enhance the developed system model for producing timetable and scheduling for various purposes like examinations, bus schedules, and many more.

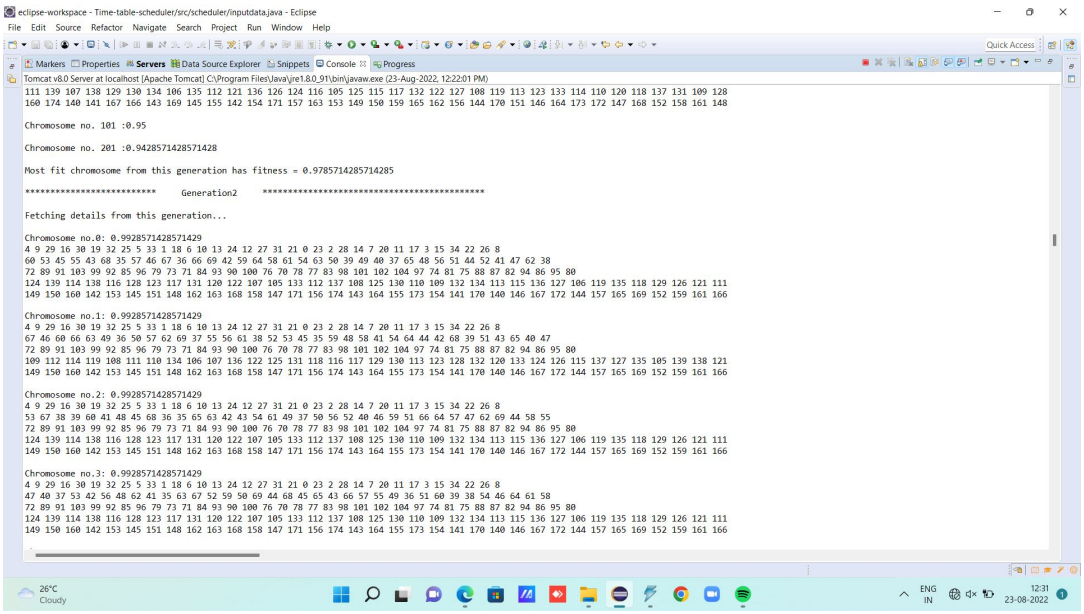

Figure 3. Genotype output on console determining fitness

REFERENCES

[1] Dipti Srinivasan Tian Hou Seow Jian Xin Xu "Automated timetable generation using multiple context reasoning for university models", IEEE conference (2002).

[2] Sadaf N. Jat, Shengxiang Yang "A mimetic algorithm for university course timetabling problem", 20th IEEE International conference on tools with artificial intelligence (2008).

[3] K. Nguyen, D. Nguyen, K. Trieu, and N. Tran, "Automating a real-world university timetabling problem with Tabu search algorithm", in 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), (2010).

[4] A. Azlan and N. M. Hussin, "Implementing graph coloring heuristic in construction phase of curriculum-based course timetabling problem", in 2013 IEEE Symposium on Computers & Informatics (ISCI), (2013).

[5] R. E. Febrita and W. F. Mahmudy, "Modified genetic algorithm for high school time-table scheduling with fuzzy time window", in 2017 International Conference on Sustainable Information Engineering and Technology (SIET), (2017).

[6] T. Elsaka, "Autonomous generation of conflict-free examination timetable using constraint satisfaction modelling", in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), (2017).

[7] K. Y. Junn, J. H. Obit, and R. Alfred, "The study of genetic algorithm approach to solving university course timetabling problem", in Lecture Notes in Electrical Engineering, Singapore: Springer Singapore, (2018), pp. 454–463.

[8] E. Psarra and D. Apostolou, "Timetable scheduling using a hybrid particle swarm optimization with local search approach", in 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), (2019).

[9] Sourabh Katoch & Sumit Singh Chauhan & Vijay Kumar, "A review on genetic algorithm: past, present, and future", Multimedia Tools and Applications, Springer (2021).

[10] L V Stepanov et al, "Mathematical modeling method based on genetic algorithm and its applications", Journal of Physics: Conference Series (2019).

[11] Peter Brucker, "Scheduling and constraint propagation", November (2002).

[12] David E. Goldberg "Genetic Algorithm in Search, Optimization and Machine Learning", (1989).

[13] John McCall, "Genetic algorithms for modelling and optimization", Journal of Computational and Applied Mathematics 184 (2005).

[14] B. Shirazi, H. Fazlollahatabar and D. Shafiei, "A Genetic Approach to Optimize Mathematical Model of Faciliies Relocation Problem in Supply Chain", Jouranl of Applied Sciences (2008).

[15] Abdelghany A, Abdelghany K, Azadian F "Airline flight schedule planning under competition", Comput Oper Res 87:20–39 (2017).

[16] Arkhipov DI, Wu D, Wu T, Regan AC "A parallel genetic algorithm framework for transportation planning and logistics management", IEEE Access 8:106506–106515 (2020).

[17] Baker JE, Grefenstette J "Proceedings of the first international conference on genetic algorithms and their applications", Taylor and Francis, Hoboken, pp 101–105 (2014).