INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

Timetable Generation– An optimal solution to the multi-constrained problem

S. Suryaa Charan
suryaacharan@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Sai Theja K.
saitheja.messi@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Jagadeesh
mjagadeesh5589@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Manikkannan
manikkannan@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

ABSTRACT

Timetabling is the appointing of an occasion to a specific timeslot in a timetable. Timetabling turns into an issue when the allocating task turns out to be difficult to be inferred where certain particular prerequisites should be satisfied. Genetic Algorithm (GA) develop as one mechanization timetabling strategy to take care of timetabling issue via seeking arrangement in multi-indicates and the capacity to refine the current answer for a superior arrangement. Genetic algorithm is a met heuristic that copies the method of natural selection. It might be performed in multiple different ways with different types but it will all follow the same concept. This research aims to create artificial intelligence through the use of an evolutionary algorithm, specifically genetic algorithm combined with adaptive and elitist traits that can generate a university schedule timetable with the goal of generating a valid and as optimal as a possible solution with certain constraints.

Keywords— *Timetabling, Multi-constrained problem, Soft computing technique, Scheduling problem, Genetic Algorithm, NP-Hard problem, Automation.*

1. INTRODUCTION

Timetable scheduling the, process of creating timetables that fit the constraint of the scenario. It is used in a magnitude of the industry from scheduling transportations up to creating a complex schedule for highly optimized automated factories. Majority of small scale scheduling is done manually while larger operations require computer-assisted scheduling. In spite of the fact that dominant part university association work has been by computers, the address timetable planning is still regularly done by hand because of its characteristic difficulties.

Lecture timetable preparation demands significant time and efforts. The manual timetable planning is a restriction satisfaction issue in which we find an outcome that satisfies the given arrangement of requirements. There have been various methodologies made in the before period to the trouble of building timetables for universities and schools. Timetabling issues might be explained by assorted techniques acquired from task concentrate, for example, genetic algorithm, simulated annealing, graph colouring, genetic algorithms, local search measures such as tabu search or from backtracking based requirement fulfilment. In our undertaking, timetable issue is defined as a limitation satisfaction issue and we proposed a reasonable timetable calculation which is fit for dealing with both hard and delicate requirements. It is a finished time table answer for universities which help to conquer the difficulties in physically developing the timetable.

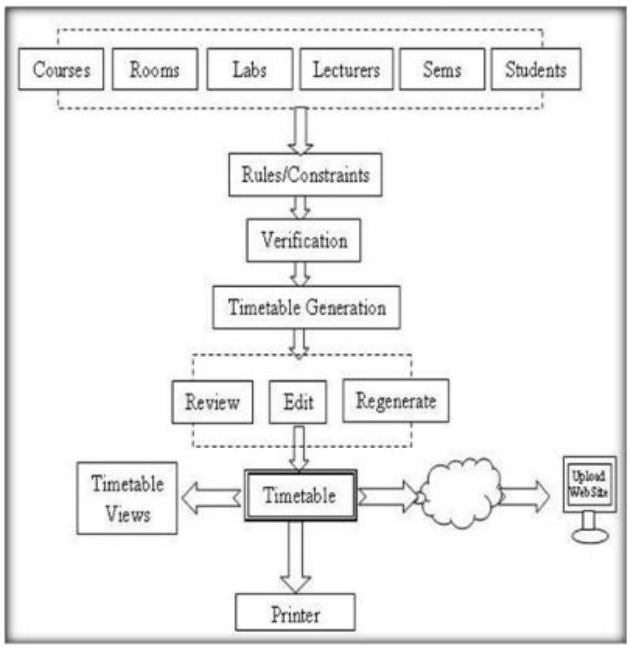

Fig. 1: General view of timetable

2. ALGORITHM

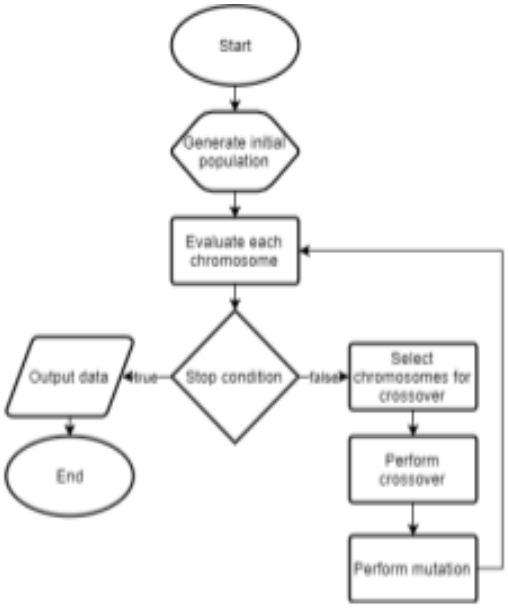

Fig. 2: Basic genetic algorithm process

2.1 Timetabling

It is expansive and very obliged, yet overall the issue contrasts significantly for various universities and learning organizations. It is difficult to compose an all-inclusive plan, fitting for all conceivable timetable issues. Despite the fact that manual making of the timetable is continued, it is as yet all-inclusive, on account of the absence of reasonable PC programs.

There exists a lot of diverse timetable problems such as:

- University Timetable
- Exam Timetable
- School Timetable
- Sports Timetable

In addition, there exists a lot of critical thinking methods, which commonly utilize the idea of optimization algorithms such as genetic algorithms, tabu search, Back-tracking, Simulated Annealing, Constrained logic Programming.

2.2 Genetic Algorithm

Genetic Algorithms - GA was imaginary by John Holland and has described this thought in his book "Adaptation in natural and artificial systems" in the year 1975. Genetic algorithm is a metaheuristic propelled by the methodology of regular choice that has a place with the greater class of evolutionary algorithms - EA. Genetic Algorithms are persuaded by Darwin's evolution theory and hypothesis. GA comes beneath the class of Evolutionary calculations that utilize the guideline of normal accumulation to build up a lot of arrangement towards the best outcome. It is a search heuristic which creates answers for enhancement issues utilizing method inspired by normal development like transformation and mutation, legacy inheritance, hybrid crossover and choice selection.

2.3 Genetic Algorithm Operators

2.3.1 Chromosome representation: Genetic material is a collection of parameters which be an arranged outcome to the issue that the algorithm seeks to answer. The chromosome in the genes can be represented as an easy string. The fitness of a chromosome is determined with its capacity to address and solve the given problem.

2.3.2 Initial population: The initial phase in the execution of a genetic algorithm is the creation of an underlying initial populace. Every individual from this populace encodes a potential answer for an issue. Every unit is assessed and appointed wellness esteem as indicated by the fitness function. It has been recognized that on the off chance that the underlying populace to the GA is great, at that point the calculation has an upgraded choice of finding a decent outcome and if the underlying supply of development squares isn't sufficiently expansive or adequate, at that point it would be hard for the calculation to locate a decent outcome.

2.3.3 Selection of good populace: This administrator chooses chromosomes in the populace for proliferation. The fitter the chromosome, the more occasions it is probably going to be picked to repeat. Amid each progressive creation, a part of the available populace is chosen to select another generation. Singular arrangements are picked through a fitness based procedure, where fitter outcome are generally bound to be picked.

2.3.4 Crossover: Crossover is a hereditary administrator used to differ the programming of a chromosome starting with one creation then onto the next. It is parallel to generation and natural hybrid, whereupon genetic algorithms are based. A crossover takes more than one parent arrangements and delivering an offspring solution. There are techniques for collection of the chromosomes. Crossover subjectively trades the subsequences when that locus between two chromosomes to make two offspring's. The crossover operator roughly does as it is a natural recombining between two single chromosomes.

2.3.5 Mutation: The mutation is utilized to continue hereditary variety from one production of a populace of GA chromosomes to the following. It is very similar to the mechanism of natural mutation. Alteration also knows as mutation changes one or more than one of the genetic values in the chromosome from its beginning circumstance. In mutation, the result may alter totally from the previous result. Hence Genetic algorithm can come to better performing result by using this process. Transformation can occur at each piece position in a string with some plausibility, normally little.

2.3.6 Fitness function: The fitness function is portrayed over the hereditary representation and systems the nature of the result. The fitness function perpetually issues dependent. Specifically, in the fields of genetic programming and hereditary calculations, each plan result. After each round of testing, the musing is to evacuate then most exceedingly terrible plan arrangement and solution.

The fitness function formula is as below:

$$F(x) = \sum_{i=1\ldots6} w_i^s \times P_i^{soft}(x) + \sum_{i=3,4,6} w_i^h \times P_i^{hard}(x) \qquad (1)$$

If fitness value =0, stop generation.
If feasible but fitness >0, user can stop generation.

2.3.7 Heuristics: It finds out a solution amongst all possible ones, but they do not prove that the most excellent will be found, they may be thought as about and not correct algorithms. These algorithms, usually find a solution close to the most excellent and they find it quick and merely. These algorithms can be correct, that is they convert and find the best solution, but the algorithm is stationary called heuristic until this best solution is proven to be the best.

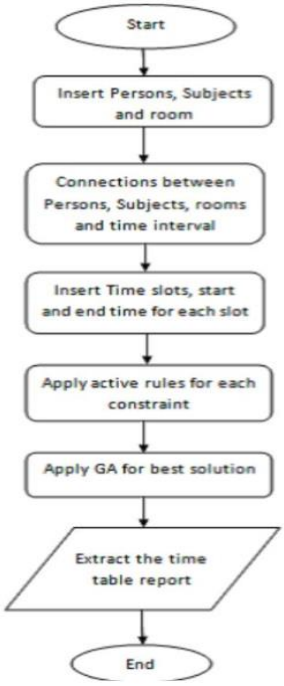

Fig. 2: Flow chart

3. DESIGN

3.1 Use case diagram

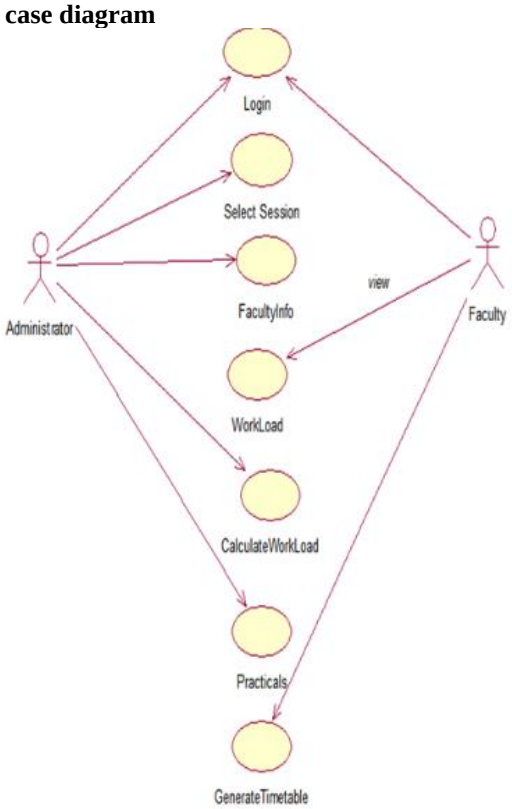

Fig. 3: Case diagram

The use case consists of a login, select session, faculty info, workload, calculate workload, practical hours, general timetable. Administrator has authorized to access all the attribute and faculty can access login workload, generate timetable. A student can view the time table. The head of the department (HOD) or the Dean of the college holds the admin privilege to make changes. The admin can add or modify the list of subjects, its code, and credits. The admin also manages the list of faculties, their designation, subjects they undertake. Preferential soft constraints can also be mentioned in the program in terms of faculty availability and preferred subject hours. Input can also be fed in via an XLS file.

3.2 Class diagram

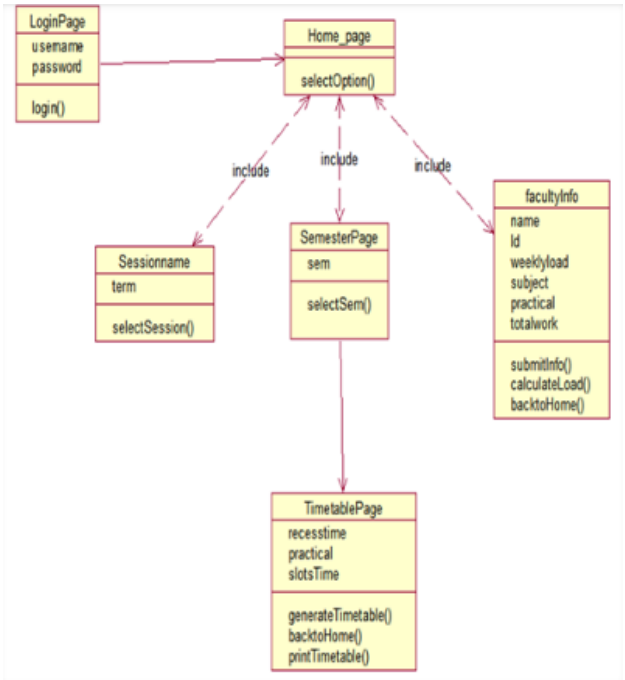

Fig. 4: Class diagram

Class diagram consist 6 classes.

- LoginPageContain attributes such as username and password and operation for allowing authenticate user to get login ().
- HomePageThis class contain the selectOption () operation for selecting select option.
- SessionnameThis class contain attributes term and operation selectsession ().
- SemesterPageThis class contain attributes sem and operation selectsem ().
- FacultyThis class contain attributes name, id weekly load, subject, practical, totalwork and operation submit info (), calculateworkload () backtohome ().
- TimeTablePage this class contain attributes recess time, practical, slotsTime and operation backtohomepage (), GenrateTimeTable (), printTimetable () .This a main class of our project.

4. CONSTRAINTS

There is an assortment of imperatives to be fulfilled at an opportunity to instantiate factors about availabilities and study halls. The imperatives can be ordered into Hard and Soft Constraints.

4.1 Hard constraints

A timetable which breaks a hard requirement is definitely not a practical arrangement. Hard limitations contain "First Order Conflicts",

- A study hall isn't doled out to more than one instructor in the same slot.
- An educator can't educate in more than one class in the same hour.
- Courses for the comparative year-session understudies of an office can't occur at the same time.

4.2 Soft constraints

Soft constraints are less noteworthy than hard constraints, and it is regularly impractical to abstain from breaking probably some of them. The timetable is utilitarian and can be utilized to a limit level to which a timetable has abused its soft constraints.

Hard Constraint	Medium Constraint	Soft Constraint
Instructors teach one class at a time	Sections' subjects are placed on the schedule	Students should have only 30 minutes break for every two hours of session per day
Instructors teach at their available schedule	Sections should have at least 30 minutes vacant time between 11:00 AM to 1:00 PM for lunch break (Optional)	Subjects that are divided should follow the two defined meeting pattern which is "MWF" and "TTH"

Fig. 5: Classification of constraints

5. REFERENCES

[1] Deshkar, M., Mayur kale, M. B., & Ghom, A. (March 2016). Time Table at a Click. International research journal of engineering and technology.

[2] Schaerf, A., "A survey of automatedtimetabling" Artificial-Intelligence Review, No. 13, 1999, pp. 87-127.

[3] A.B. Luis E., S.S. Sancho, and O.G. Emilio G. etc. A hybrid grouping genetic algorithm for assigning students to preferred laboratory groups. Expert Systems with Applications, vol. 36, No. 3, PART 2, 2009, pp. 7234–7241.

[4] De Werra D., "An introduction to timetabling", European Journal of Operation Research", Vol. 19, 1985, pp. 151-162.

