Timetable Generator

Swapnil Markal, Surekha Ghorpade, Diksha Chalke

[1](Computer Science, Bharati Vidyapeeth College of Engineering/ Mumbai University, India)
[2](Computer Science, Bharati Vidyapeeth College of Engineering/ Mumbai University, India)
[3](Computer Science, Bharati Vidyapeeth College of Engineering/ Mumbai University, India)

Abstract: *The manual system of generating a timetable is a tedious job for educationalist in colleges with large number of students with respect to time and manpower which usual ends up with various classes clashing either or multiple lectures assigned to the same professor. Providing an automatic timetable generator will help to generate exam and class timetable automatically. The system will help to generate it automatically and will also help to save time. It avoids the complexity of the manual work for setting and managing timetable. The aim here is to develop a simple, efficient and easily accessible application, which will ease the process of generation and distribution of timetable. The main algorithm used in timetable generation is Genetic algorithm. It helps to produce a best schedule by regulating all the rules. There is no need for faculty to worry about time clashes. This system will be having easy to use, interactive and less complex interface with inputs like teacher name, the data for the rooms, and data for the labs and the data for the subject. The system will have a well-designed database to store all the information entered as input. The system will have algorithm to process all the data present in the database considering hard and soft constraints. An approach of a timetable generator by applying Genetic algorithm in order to obtain optimal solution. Instead of tedious paperwork, the students and faculty can view the timetable with a quick turnaround. This system is user friendly and helps to generate a timetable efficiently.*

Key Word*: Chromosomes, Genes, Evolutionary Algorithm, Genetic Algorithm, Mutation, Soft Computing.*

Date of Submission: 15-04-2020   Date of Acceptance: 30-04-2020

I. Introduction

It's important for humans to schedule their work efficiently. Without a proper timetable daily routine gets messy. Proper goal setting gets difficult. In Educational institutes, it is important to have a proper timetable scheduled. Every semester there is a limited time and every subject need different amount of time to complete. Therefore, it is important to schedule an efficient timetable so that right amount of time is allotted to the subjects. Person responsible for this task manually may have to do lots of permutation and combination to reach a perfect timetable. This may result into consumption a lot of time and energy. Also, this perfectly scheduled timetable may have to be changed any time for many reasons like there can be a change in the number of teachers, number of public holidays may increase and many more. There is no certainty for this timetable to be perfect for the whole semester. Now if this happens too many times, generating timetable every time may not be possible and may develop lots of confusion in daily routine. Therefore, a Timetable Generating System is very essential in educational institutes. Generating a perfect Timetable in a lesser time will get easy. All the complex permutation and combination will be performed by the system very efficiently. The person responsible for generating a timetable will not have to worry about the time clashes. The system will provide easy and less complicated interface making it easy for the user to make use of it. The user will have to provide inputs like teacher name, subject name, number of classrooms, labs, duration of the course. The system will have a well-designed database to store all the inputs by the user. This data will be analyzed by the algorithm. Teachers and student won't be required to note the timetable. They will be able to view the timetable with ease.

II. Literature Review

"Dipti Srinivasan Tian Hou Seow Jian Xin Xu" expressed that an ideal timetable which fulfills all the requirements has become a provoking errand for educational establishments to overcome. This paper presents a transformative calculation called Evolutionary Algorithm (EA) based way to deal with vigorously compelled timetabling issue. The approach uses a problem specific chromosome representation. Heuristics and context-based thinking have been utilized to produce a timetable which fulfills the greater part of the requirements and can create the timetable in attainable time. An intelligent adaptive mutation technique has been used to rush the process of generation of timetable. Yet, this framework ends up being badly arranged to actualize since it thinks about whole college issue and transformative calculation

Antariksha Bhaduri "Timetable Scheduling utilizing Genetic Artificial Immune Network" [2] in their article suggested that Scheduling is one among the significant undertakings experienced in certifiable circumstances. There are numerous spots where there is a requirement for planning the timetable, similar to personal scheduling, production scheduling, education timetable scheduling etc. Educational timetable booking is a troublesome undertaking as a result of the numerous requirements that are should have been fulfilled so as to get a doable arrangement. Educational timetable booking issue is comprehended to be NP Hard. Thus, transformative systems are wont to take care of the timetable booking issue. Hereditary Algorithms, Evolutionary Algorithms and so on have been utilized with blended achievement. Right now, have surveyed the issue of instructive timetable planning and fathoming it with Genetic Algorithm. We have additionally unraveled the issue with a mimetic half and half calculation, Genetic Artificial Immune Network (GAIN) and look at the outcome immediately got from GA. Results show that GAIN is in a situation to prevail in the ideal achievable arrangement quicker than that of GA.

Mei Rui, right now, study and subsequently the rundown of the overall issues, a scientific model for the course timetable framework is proposed. At an identical time, utilizing the example acknowledgment innovation in AI, focusing on this numerical model a substitution college course timetable framework configuration program is proposed and figured it out. This program not exclusively can well explain the deficiencies of the predominant course timetable framework, but on the other hand is basic and simple to work, has solid flexibility. [1]
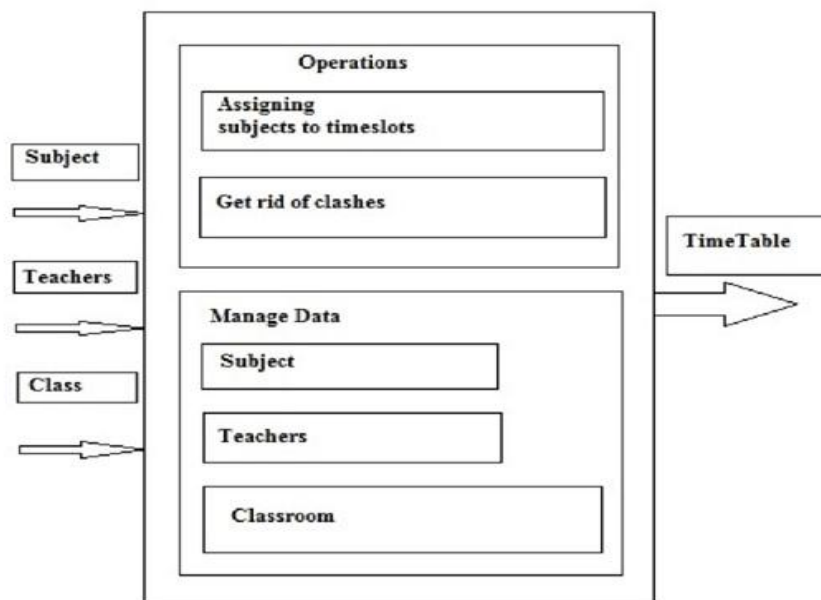
III. Architecture

Figure: Block Diagram of Timetable Generator.

The figure represents the Timetable Generator. Consisting of input module and required Timetable as the output.
A. Input Data:
Input Data can be described as different inputs that are fed to the system by the user.
They are:
1. Teachers: It will contain data related to teacher like name and subject taught by him/her.
2. Subject: It will contain data related to the subject like name and time required to complete the subject.
3. Rooms: It will contain data related to classrooms and labs.
4. Time interval: It will contain the duration of lectures and total lectures to be conducted in a day.

B. Constraints:
Constraints are of 3 types:
1. Validity violation constraints
2. Hard Constraints
3. Soft Constraints

1. Validity Violation Constraints:

These are some constraints which need to be followed:

- No teacher can be allotted two lectures at the same time.
- Two different lectures cannot be allotted a same class at the same time.
- Fixed Time slots.

2. Hard Constraints:

These are some constraints which cannot be ignored:

- Every subject lecture must be allotted only once in a day.
- Classrooms or labs cannot be double booked.
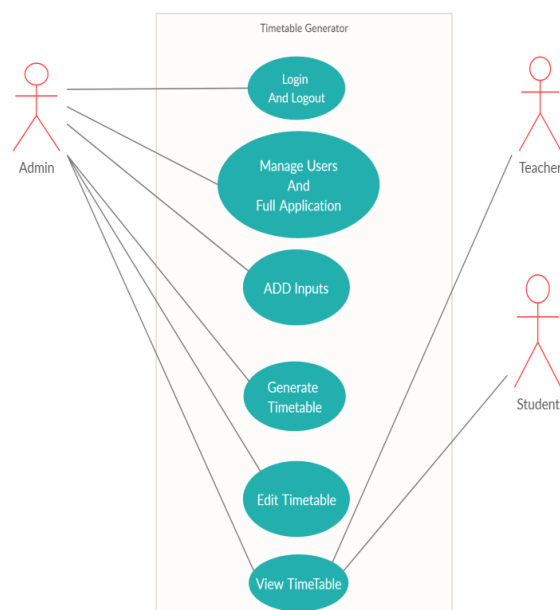
3. Soft Constraints:

These are some constraints that are not compulsory. More the number of these constraints are followed, the timetable generated can be said as perfect timetable

- No continuous lecture for the teacher in the class.
- A teacher should not have consecutive lectures.
- Fixed slot for a subject.

IV. Workflow
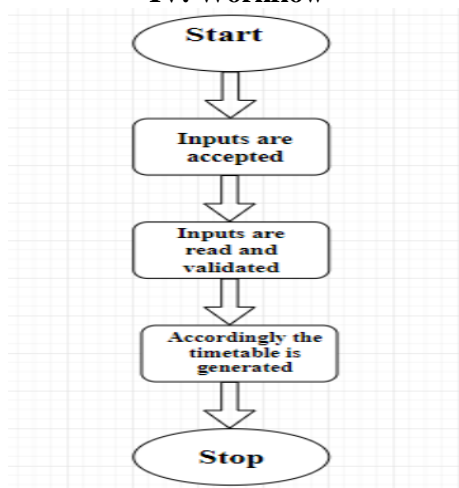
V. Use-Case Diagram

VI. Methodology

Genetic Algorithm (GA) is a search-based optimization technique supported the principles of Genetics and survival. It's frequently wont to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to unravel. It's frequently used to solve optimization problems, in research, and in machine learning. Nature has always been an excellent source of inspiration to all or any mankind. Genetic Algorithms (GA's) are search based algorithms supported the concepts of survival and genetics. GA's may be a subset of a way larger branch of computation referred to as Evolutionary Computation. John Holland along with his students and colleagues at the University of Michigan developed Genetic Algorithm. In GA's, we've a pool or a population of possible solutions to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and therefore the process is repeated over various generations. Everyone (or candidate solution) is assigned a fitness value (based on its objective function value) and therefore the fitter individuals are given a better chance to mate and yield more "fitter" individuals. This is often in line with the Darwinian Theory of "Survival of the Fittest". During this way we keep "evolving" better individuals or solutions over generations, till we reach a stopping criterion. Genetic Algorithms generate random solutions, but they perform far better than random local search, as they exploit historical information also.

Search space:

The population of individuals are maintained within search space. Each individual represents a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus, a chromosome (individual) is composed of several genes (variable components).
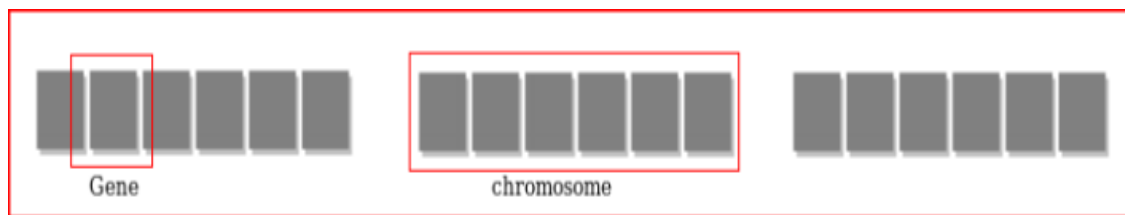
A Fitness Score is given to everybody who shows the capacity of a person to "contend". One with ideal or close to ideal wellness score are chosen. The GAs keeps up the number of inhabitants in n people (chromosome/arrangements) alongside their wellness scores. The people having better wellness scores are given more opportunity to repeat than others. The people with better wellness scores are chosen who mate and produce better posterity by consolidating chromosomes of guardians. The populace size is static, so the room must be made for fresh introductions. Along these lines, a few people kick the bucket and get supplanted by fresh introductions in the end making new age whenever all the mating chance of the old populace is depleted. It is trusted that over progressive ages better arrangements will show up while least beyond words. Each new age has on normal progressively "better qualities" than the individual (arrangement) of past ages. In this way, each new age has better "halfway arrangements" than past ages. When the posterity's delivered having no noteworthy contrast than posterity created by past populaces, the populace is met. The algorithm is said to be changed over to a lot of answers for the problem.[8] Operators of Genetic Algorithms Once the underlying age is made, the algorithm advances the generation utilizing following operators –

1) Selection Operator: The thought is to offer inclination to the people with great wellness scores and permit them to pass their qualities to the progressive ages.

2) Crossover Operator: This speaks to mating between people. Two people are chosen utilizing selection operator and hybrid locales are picked haphazardly. At that point the qualities at these hybrid locales are traded therefore making a totally new individual (posterity). For instance –
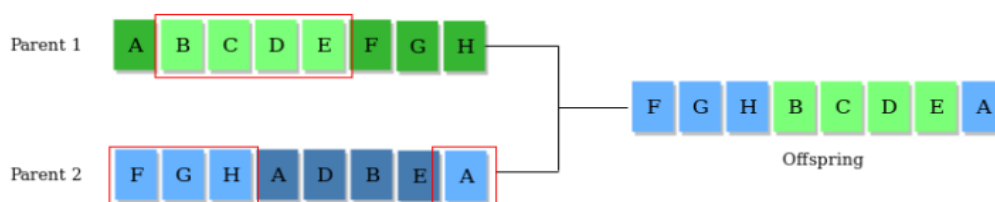
3) Mutation Operator: The key thought is to embed arbitrary qualities in posterity to keep up the decent variety in populace to maintain a strategic distance from the untimely assembly. For instance –

The entire calculation can be condensed as –
1) Randomly instate populace's p
2) Determine wellness of populace
3) Until union rehash:
a) Select guardians from populace
b) Crossover and create new populace
c) Perform transformation on new populace
d) Calculate qualification for new populace.
Given an objective string, the objective is to deliver target string beginning from an arbitrary string of a similar length. In the accompanying usage, following analogies are made –
Characters A-Z, a-z, 0-9 and other special symbols are considered as qualities A string produced by these characters is considered as chromosome/arrangement/Individual Fitness score is the quantity of characters which contrast from characters in target string at a specific file. Along these lines, the individual having lower wellness esteem is given more inclination.

VII. Conclusion

The Timetable Generator will ease the efforts of manually generating the timetable. The process will be easier and more efficient. It will reduce the time and complexity involved in manual generation of timetable.

References

[1]. Mei Rue Computer and Automation Engineering (ICCAE), 2010   The 2nd International Conference on (Volume:4)
[2]. Antariksha Bhaduri "University timetable scheduling using genetic algorithm". Advances in recent technologies in Communication and Computing, 2009. ARTCom '09. International Conference
[3]. Adithya R Pai, Ashwitha S, Raksha Shetty "Automated college timetable generator" .2018 International Journal of Scientific & Engineering Research Volume 9.
[4]. Vikas Gawade, Kunal Deokar, Prasanna Retherekar, Shrinath Bhegade "Timetable Generation". 2016 Internation Engineering Research Journal Volume 2
[5]. Saritha M, Pranav Kiran Vaze, Pradeep, Mahesh N R "Automatic Timetable Generator". 2017 International Journal of Advanced Research in Computer Science and Software Engineering.
[6]. Anuja Chowdhary, Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, Dinesh Gawande "Timetable Generation System". 2014 International Journal of Computer Science and Mobile Computing.
[7]. Parkavi A "A Study on Automatic Timetable Generator". 2018 International Journal of Innovative Research & Growth.
[8]. www.geeksforgeeks.com