

Automatic Timetable Generation using Genetic Algorithm.

Shraddha Shinde, Saraswati Gurav, Sneha karme.

Abstract— The implementation of a computer program which employs Genetic Algorithm (GAs) in the quest for a optimal lecture timetable generator. GA theory is covered with emphasis is on less fully encoded systems employing non-genetic operators. The field of Automated Timetabling is also explored. A timetable is explained as essentially, a schedule with constraints placed upon it. The program written in java that has a good object oriented to do it, and it has the special libraries to deal with genetic algorithm which will be used for the implementation. In simplified university timetable problem it consistently evolves constraint violation free timetables. The effects of altered mutation rate and population size are tested. It is seen that the GA could be improved by the further incorporation of repair strategies, and is readily scalable to the complete timetabling problem.

Index Terms— Minimum 7 keywords are mandatory, Keywords should closely reflect the topic and should optimally characterize the paper. Use about four key words or phrases in alphabetical order, separated by commas.

1 INTRODUCTION

The Automatic Timetable Generator is a Java based software used to generate timetable automatically. Currently timetable is managed manually. It will help to manage all the periods automatically and also will be helpful for faculty to get timetable in their phone by using application. It will also manage timetable when any teacher is absent, late coming or early going. Maximum and minimum workload for a Faculty for a day, week and month will be specified for the efficient generation of timetable. By using this software users can apply for leave by providing leave required date, reason and also with substitute faculty. When selecting a faculty as substitute it allows to view timetable of that faculty for ensure that the faculty is free at that particular period. Substitute can approve or reject request. Principal can also view the request and send by faculty and can also view substitute response. Principal can approve reject request. It is a comprehensive timetable management solutions for colleges which help to overcome the challenges in manually setting the timetable. By using this software it will be very easy for faculty to get timetable in their phones.

2 PROJECT STATEMENT

Normally timetable generation done manually. As we know all institutions\organizations have it's own timeable, managing and maintaining these will not be difficult. Considering workload with this scheduling will make it more complex. As mentioned, when timetable generation is being done, it should consider the maximum and minimum workload that is in a college. In those cases timetable generation will become more complex. Also it is time consuming process.

3 WHAT IS TO BE DEVELOPED?

Automatic Timetable is a java based software is used to generate timetable Automatically. Will help you to manage all the periods Automatically and also will be helpful for faculty who

will get timetable in their phone as a notification. It will also manage timetable when any teacher is absent is late coming and early going. Proposed system will help to generate it automatically also helps to save time. There is no need for faculty to worry about about period details and maximum workload. By using this software user can apply for leave required date, reason and also with substitute faculty. When selecting a faculty as substitute it allows viewing timetable of that faculty for ensure that the faculty is free at that particular period. Substitute can approve or reject request. Principal can also view the request send by faculty can also view substitute response. Principal can approved\reject request. It is a comprehensive timetable management solutions for Colleges which help to overcome the challenges in current system.

4 FEASIBILITY ANALYSIS

A feasibility study was an evaluation of a proposal designed to determine the difficulty in carrying out a design task. Generally, a feasibility study precedes technical development and project implementation.

4.1 ECONOMIC FEASIBILITY

To develop the proposed system, it needs no extra facilities and devices. All dependencies are satisfied from the open source project. All tools used are free, open source and programming language is JSP and hence its development is economically.

4.2 TECHNICAL FEASIBILITY

Proposed system is technically feasible because the proposed system requires only those H\W and S\W tools that are available in the system. It requires the installation of JSP and

MYSQL which can be done for free . more over expandibility will be maintained in the system. New models can be added later on the application, if required in the future. Additionally application will have User friendly Forms and Screens.

4.3 BEHAVIORAL FEASIBILITY

Behavioral feasibility determines how much efforts will go in the proposed information system, and in educating and training the users on the new system,. Since the user interface is very simple and easily understandable, no training is required for using the software.

5 GENETIC ALGORITHM

1. An algorithm is a set of instructions that is repeated to solve a problem.
2. Genetic algoritms follow the idea of SURVIVAL OF THE FITTEST- Better and better solutions evolve from previous generations until a near optimal solution is obtained.
3. Also known as evolutionary algorithms,genetic algoritms demonstrate self organization and adaption similar to the way that the fittest biological organism survive and reproduce.
4. A genetic algorithm is an interative procedure that represents its candidate solutions as strings of genes called chromosomes.
Generally applied to spaces which are too large.
5. Genetic algorithms are often used to improve the performance of other AI methods such as expert system or natural selection.
6. The method learns by producing offspring that are better and better as measured by a fitness function, which is measure of the objective to be obtained. (maximum or minimum)

Each user has their own functionalities as follows.

Function: subject allocation

Input : subject, faculty and semester

Output : timetable

Logic:

Step 1: Read subject, faculty and semester from the tables subjects, faculty and semester respectively.

Step 2: Validate the details and processing is done

Step 3: subject allocated for faculty successfully in the table named Timetable.

Function: view Timetable

Input : Subject, Semester, Faculty

Output : Timetable

Logic:

Step 1: Read date, reason and substitute faculty from the table fac_leave

Step 2: Get substitute status from fac_leave

Step 3: Request successfully sent (Approve / Reject)

If faculty needs to take a leave the he/she can apply for leave with setting Substitutes. Here Date at which leave is required, reason and substitute faculty should be specified.

Function: Substitute

Request input: Request

Output: Approved/Reject

Logic:

Step 1: Read the information and apply for leave with providing date, reason and substitute faculty in the table fac_leave

Step 2: substitute request successfully sent (Approve/Reject)

Function: Grant or Deny

Input : Request

Output : Reply message send Approve/Reject

Logic:

Step 1: Access substitute request send by the faculty from fac_leave

Step2: Reply for Substitute request successfully sent (Approve/Reject)

Function: Grant leave

6 CODING AND TEMPLATES

The system contains 3 users, Faculty, Principal and Admin.

Input : Request
Output : Reply message send Approve/Reject

Logic:

Step 1: Access leave request send by the faculty from fac_leave

Step 2: Reply for Request for leave successfully sent (approve/reject)

Function: Subject allocation
Input: subject, faculty and semester
Output: timetable

Logic:

Step 1: Read subject, faculty and semester from the tables
Subject, faculty and semester respectively.

Step 2: Validate the details and processing is done

Step 3: Timetable generated successfully.

7 ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to everyone who helped and guided us to complete the project successfully. First of all, we are so much grateful to the valuable effort made through during this course, in order to achieve a great experience of outcome a successful software project at the end. We are also grateful to our family and friends, for supporting and encouraging completing this project successfully. Finally, we would like to thank all the colleagues who have been with us in all difficult times with suggestion and supportive words which carry us to make this project a reality.

9 LITERATURE SURVEY

To develop software which help to generate Timetable

For an Institution automatically. By looking at the existing system we can understand that timetable generation is done manually. Manually adjust the timetable when any of the faculty is absent, and this is the big challenge of Automatic Timetable Generator that managing the timetable automatically when any of the faculty is absent. As we know all institutions/organization have it's own timetable, managing and maintaining this will not be difficult. Considering workload with these scheduling will make it more complex. As mentioned, when Timetable Generation is being done, it should consider the maximum and minimum workload that is in a college. In those cases, Timetable Generation will become more complex. Also, it is a time consuming process.

8 CONCLUSION

We have shown that a genetic algorithm approach is very effective and useful on the lecture timetabling problems. Using the method we have described and shown a great potential for leading timetable in future which are fairer to students. The framework seems directly applicable to a very wide variety of other timetabling problem. For example, experimental result shows that a key aspect towards its success is the employment of the mutation operator described. The GA in timetabling framework has been shown to be successful on several real problem 'University Department size', and so it seems we can justify the expectation for it to work very well on other problems of similar size and nature. That is, there is no reason to suspect that there is anything particularly easy about the problem it was tested on. In comparison to other real problems. Much work remains to do to see how performances scale to larger and otherwise different kinds of timetabling problems.

9 REFERENCES

- [1] Alberto colorni, Marco Dorigo, Vittoria Maniezzo, A GENETIC ALGORITHM TO SOLVE THE TIMETABLE PROBLEM, Centre for Emergent Computing, Napier University, Edinburgh EH10N 5DT, UK 2000.
- [2] 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 Automata.