Exam Cell Automation System and Timetable Generator

Namrata Bodas[1], Jhanvi Shah[2], Yash Shah[3], Aishwarya Sontakke[4], Dnyaneshwar Dhangar[5]
[1,2,3,4] Student, Computer Engineering Department, Rajiv Gandhi Institute of Technology, Versova, Mumbai, India
[5]Professor, Computer Engineering Department, Rajiv Gandhi Institute of Technology, Versova, Mumbai, India

Abstract: Exam Cell Automation System is developed for the college or institute to computerize the traditional way of conducting exams and to simplify examination hall allotment and seating arrangement. Also, automating the timetable generating process will help get rid of various anomalies caused due to human error. Mostly staff faces many problems in assigning exam halls depending upon the availability of classes, capacity of classes, student details, allotment of duties to the respective teachers, etc. And the manual system of preparing time table in colleges is very time consuming and usually ends up with various classes clashing either at same room or with same teachers having more than one class at a time. To overcome all these problems, an automated system is proposed.

IndexTerms - Exam Cell Automation System, Timetable Generation System, Constraints, Heuristic algorithm, optimal solution.

I. Introduction

Exam cell allotment is a process of complete allotment of students and faculty to their respective classrooms for given examination. It involves various sub-processes like managing the time allotment for exams of different courses from different academic years on the same day (in case of internal exams) or over a given period and scheduling the faculty according to their workload, duties, etc. A college timetable is a temporal arrangement of a set of lectures and classrooms in which all given constraints are satisfied. Creating both of these systems manually is complex and time-consuming process. The proposed software is used to overcome the entire problem which they are faced and making complete automation of manual system to computerized system.

1.1 Purpose

A completely automated output will be generated by the final system which will save a lot of time and effort of an institute administration. The system should be able to handle all the user defined constraints and it should achieve ease of use for user of system so that he/she can make automatic time table. The system should focus on the optimized use of resources. We also aim to make the system generic so that it can work equally well for different institutes and colleges.

1.2 Scope

The scope of the project is the system on which the software is installed, i.e. the project is developed as a third-party general-purpose software, and it will work for a particular institute which can efficiently generate optimal solutions.

II. System Analysis

2.1 Existing system

Existing system is based on manual paperwork and manual calculations. The amount of work makes this process very slow and tedious, making it inefficient. The system requires maintaining a lot of documents. Handling of such important documents is again a tedious work. Also, retrieval of important facts and statistics will consume time. Since, there is a lot of manual work involved in current system, mistake in one detail can lead to wrong generation of page. This system faces a huge problem as there is no proper collection of requirements. This system is to enhance manual work and also more energy is wasted to allocate the seating arrangement.

2.2 Disadvantages of Existing System

- The retrieval of data is very slow and data is not maintained efficiently.
- More calculations are required to generate the report so it is generated at the end of the session.
- All calculations for generating reports are done manually which may lead to errors.
- Existing system requires lots of paper work. Loss of even single register/record can lead to difficult situation because all the papers are needed to generate the reports.
- Report generation is very time consuming as the work done is manual and reports cannot be generated in the middle of the session.

III. Proposed System

Enterprise-oriented software is to be designed to automate both exam hall allocation and generate timetable. The system will take various inputs like information of students, number of available classrooms, faculty, courses, etc. Depending upon the inputs it will generate a possible output, making optimal utilization of all resources that will best suit any of constraints or college rules. The project showcases an automated system which ensures the reduction of the tediousness, more effective work and systematic management. This system will help the college in saving the extra time spent in manual work, avoiding mistakes due to human error, will increase efficiency and save time, and will allows neat handling of data instead of error prone records.

The proposed system includes

- Interface for input
- Database Capabilities
- Processing Capabilities
- Search Panel

3.1 Characteristics of Proposed System

- User friendly: The retrieval and storage of data is fast and data is maintained efficiently because the proposed system is user friendly. The GUI is provided in the proposed system, that enables user to deal with the system very easily.
- Easy report generation: Reports can be easily generated in the proposed system. So that the user can generate the report as per the requirement or in the middle of the semester.
- Less paper work: The proposed system needs little paper work. All the data is put into the computer and reports can be generated through computers. Moreover, work becomes easy because there isn't any need to keep data on paper.
- Computer operator control: Computer operated control will be present so that there is no chance of errors. Moreover, storing and retrieving of information is easy so work can be done quickly and in time.
- Boosts enterprise accessibility.

IV. Architecture Diagrams

This project will take different inputs from the users like number and information of Teachers, Courses, Semesters, etc. and Timeslots also rules, facts and constraints, that are stored in XML based knowledge base. This will be the input to timetable generator algorithm on the server. After the representation of our knowledge base is done, designing of the timetabling algorithm will be done. The flow for the timetable management system part of the software is shown in the diagram below:

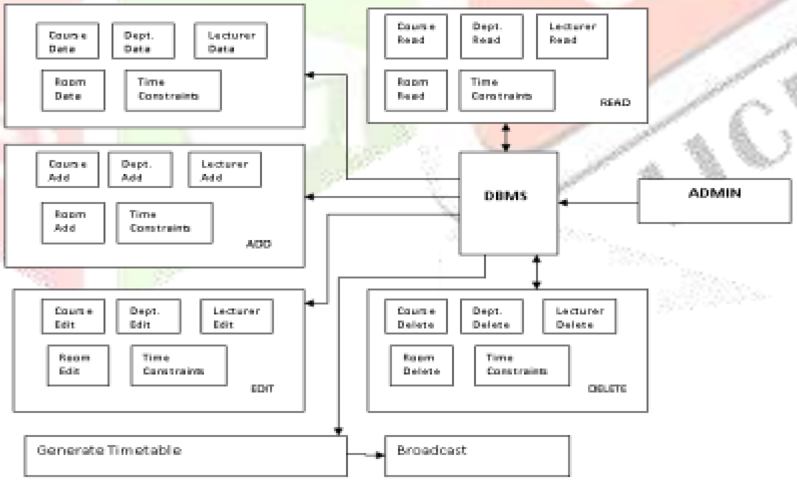

Figure 1: Timetable Generation

The input data contains:

1. Professor: Data describes the name of lecturers along with their identification number.

2. Subject: Data describes the name of courses in the current term.

3. Room: Data describes the number of rooms and their capacity.

4. Time intervals: It indicates starting time of the lecture along with duration.

System Constraints for Timetable Generation are divided into 2 categories:

1) Hard Constraints: The timetable is subjected to the following four types of hard constraints, which must be satisfied by a solution to be considered as a valid one. A student should have only one class at a Time.

   a) A Teacher should have only one class at a time.
   b) A room should be booked only for one class at a time.
   c) Some classes require classes to have particular equipment. For instance, audio visual equipment, projectors etc.

2) Soft Constraints: These are the constraints that are of no great concern but are still taken into contemplation. They don’t need to be satisfied but the solutions are generally considered to be good if they are satisfied.
   a) Courses must be eventually distributed.
   b) Students should not have any buffer time between two classes on a single day.
   c) Scheduling of teachers should be evenly divided throughout the week.

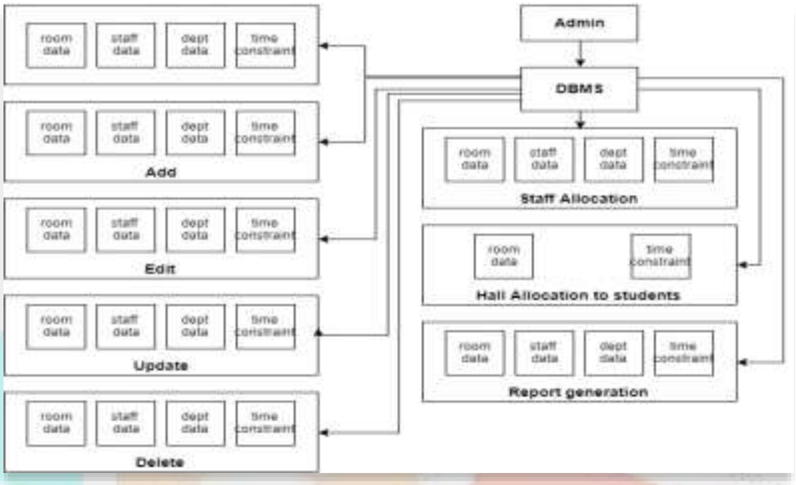

Figure 2: Exam Cell Automation System

This project provides a way to allocate exam hall for each student without clash. The system will store the list of classes and its capacity where exam can be conducted and list of students appearing the exam on each day. The system provides facility to generate report of the hall allocation.

The input data contains:

1. Professor: Data describes the name of invigilators along with their identification number.

2. Duties: Date describes the name of the invigilators along with number of duties to be assigned to them.

3. Subject: Data describes the name of courses in the current term.

4. Room: Data describes the room number and their capacity.

5. Time intervals: It indicates starting time along with duration of the exam.

6. Student: It indicates the total number of students, their roll numbers and the divisions to which they belong.

System Constraints for exam cell automation system are divided into 2 categories:

1) Hard Constraints: The exam cell automation is subjected to the following three types of hard constraints, which must be satisfied by a solution to be considered as a valid one.
   a) A student should have only one exam at a Time.
   b) A Teacher should have only one exam invigilation duty at a time.
   c) A room should be booked only for one exam at a time.

2) Soft Constraints: These are the constraints that are not the real concern but are yet taken into consideration. They don’t need to be satisfied but the solutions are generally considered to be good if they are satisfied.
   a) Course teacher should not be assigned exam invigilation duty for the same subject for the same class.
   b) Students should have some free time between two exams on a day.
   c) Scheduling of teachers should be well spread over the examination period.

V. METHODOLOGY

The search space of this problem is too vast, a lot of solutions exist in the search space and few of them are not even feasible. Feasible solutions here mean those that do not violate hard constraints as well as try to satisfy soft constraints. We choose the most appropriate one from feasible solutions. Most appropriate solutions mean those that do not violate soft constraints to a greater extent. Using Heuristic Algorithm, a number of trade-off solutions, in terms of multiple objectives of the problem, could be obtained very easily. Moreover, any of the obtained solutions would be much better than a manually prepared solution which is in use.

5.1 Heuristic Algorithm

The term heuristic is used for the algorithms which find solutions amongst all possible ones, but they don't guarantee that the best solution will be found, hence they may be considered as approximate and not accurate algorithms. Heuristic algorithms usually find a solution closest to the most preferable solution and they find it fast and easily. The method that is used for heuristic algorithm here, is greediness, but for it to be simple and quick the algorithm ignores or suppresses some of the problem's demands.

Heuristic algorithm is used, to place the activities in turn, beginning with the most difficult ones. If the algorithm is not able to find a solution, it points to the potential impossible activities, so you can correct errors. The algorithm recursively swaps activities if that is possible in order to make space for some new activity, or, in extreme cases, backtracks and switches order of evaluation. The algorithm mimics the operation of a human timetable.

Input: a set of activities A_1...A_n and the constraints.

Output: a set of times TA_1...TA_n (the time slot of each activity). The algorithm must put each activity at a time slot, respecting constraints. Each TA_i is between 0 (T_1) and max_time_slots-1 (T_m).

Constraints:

C1) Basic: a list of pairs of activities which cannot be simultaneous (for instance, A_1 and A_2, because they have the same teacher or the same students);

C2) Lots of other constraints.

VI. Software Environment

6.1 QT

Qt is a cross-platform application and graphical user interface (GUI) framework, which is used in the development of software that can run on various hardware platforms and operating systems. Qt makes it simple to develop a software with native-looking (to the OS its running on) GUIs using standard C++, which is why it is conjointly classified as a widget toolkit.

Qt offers the subsequent benefits:

- Maintaining one source tree
- Porting an application to multiple platforms through straightforward recompilation
- Broadening the audience of a product
- Creating an application that runs at native speed and with a native look and feel to the platform it is running on.

6.2 CPP

C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Stroustrup, it is also an extension of the C language. Hence it is possible to code C++ in a "C style" or "object-oriented style." In some situations, it can be coded in either manner and is so a good example of a hybrid language.

C++ could be an extremely portable language and is the language of selection for multi-device, multi-platform software development.

6.3 HTML

HTML (Hypertext Markup Language) is that the set of markup symbols or codes inserted in a file meant for display on a Web browser page. The markup tells the Web browser how to display a Web page's words and pictures for the user.

VII. Conclusion

Ultimately the result of the implementation of this project will lead to reduced workload of the staff. The result would be a fully-fledged working Automated Exam Cell Automation and Timetable Generation System. There won't be any need to use multiple different systems for different activities. The processes will be covered by the proposed system. Various slot combinations can be acquired so that another timetable is generated as of need. The project reduces time consumption and the pain in framing the timetable manually. This will reduce the tediousness of the manual processes and give a chance for efficient, flexible and automated process.

References

[1] P. Boizumault, Y. Delon, and L. Peridy, "Constraint logic programming for examination timetabling," The Journal of Logic Programming, vol. 26, no. 2, pp. 217 – 233, 1996.

[2] M. Carter, G. Laporte, and S. Lee, "Examination timetabling: Algorithmic strategies and applications," The Journal of the Operational Research Society, vol. 47, pp. 373 – 383, Mar. 1996. 1996.

[3] G. Dueck, "New optimization heuristics: The great deluge algorithm and the record-to-record travel," Journal of Computational Physics, vol. 104, pp. 86 – 92, January 1993.

[4] N. Pillay, "A developmental approach to the examination timetabling problem." http://www.cs. qub.ac.uk/itc2007/winner/bestexamsolutions/pillay.pdf, 2007.

[5] E. Burke and J. Newall, "Solving examination timetabling problems through adaption of heuristic orderings," Annals of Operations Research, vol. 129, no. 1-4, pp. 107–134, 2004.

[6] E. Burke, R. Qu, and A. Soghier, “Adaptive selection of heuristics for improving exam timetables,” Annals of Operations Research, vol. 218, no. 1, pp. 129–145, 2014.

[7] S. Rahman, A. Bargiela, E. Burke, E. Özcan, B. McCollum, and P. McMullan, “Adaptive linear combination of heuristic orderings in constructing examination timetables,” European Journal of Operational Research, vol. 232, no. 2, pp. 287–297, 2014.

[8] P. Carvalho, “Lecture notes in evolutionary computation.” Instituto Superior Técnico, November 2004.

[9] S. and Burke E.K. 2004. University Timetabling In: Leung J. (ed.) “Handbook of Scheduling: Algorithms”, “Models, and Performance Analysis.” Chapter 45. CRC Press.