Optimized Timetable Generator

- Project Proposal -

A.D.Hendawitharana

M.M.Illesinghe

M.S.J.Sumudini

Department of Computer Engineering
University of Peradeniya

Final Year Project (courses CO421 & CO425) report submitted as a requirement of the degree of
B.Sc. Eng. in Computer Engineering

March 2017

Supervisors: Mr. Sampath Deegalla (University of Peradeniya), Dr. S. P. C. Perera (University of Peradeniya)

We would like to dedicate this thesis to Mr. Sampath Deegalla (University of Peradeniya),Dr. S. P. C. Perera (University of Peradeniya)

Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is our own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgments.

A.D.Hendawitharana
M.M.Illesinghe
M.S.J.Sumudini
March 2017

Acknowledgements

Apart from the efforts of ourselves, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful startup of this project. We would like to show our greatest appreciation to Mr. Sampath Deegalla, Dr. S. P. C. Perera and Dr. Roshan Ragel for giving us a good guideline for project throughout numerous consultations.Further we would also like to expand our deepest gratitude to all those who have directly and indirectly guided us.

Abstract

Manual creation of time tables for universities is an expertise work. If there is a mistake happens that can cause many conflicts.To overcome this problem we can automate the process. The proposed system will take different inputs like number of students, lecture halls, lecturers, capacity of lecture halls and all the data and it will create feasible time tables for working days of the week, making excellent application of all resources in a way which will be best suited for the constraints. Normally time table generation face various barrier constriants of resources. This project includes algorithm which is capable of handling both hard and soft constraints more effectively. This will also show the time table related to each student and lecturer.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 What is an Optimized Timetable Generator?

Usually in most universities timetable scheduling for academics and exams is done manually. Time tables for academics are created in the beginning of each Semester. There are a lot of implementations for automated time tables. But they cannot be applicable for every university since they have their own constraints like lecture halls, time-slots, etc. Our motivation is to create an Optimized Timetable Generator to reduce the time and effort which we need for creating time tables. Furthermore we hope to develop the product in a more commercial way by adding some user friendly features.

1.2 Optimized Timetable Generator

1.2.1 Background

Time table generation is a constraint satisfaction problem. In this scenario we can define constraints as hard constraints and soft constraints. Hard constraints which is also known as mandatory constarint can be defined as the constraints which can not be violated during the implementation.

- No student can attend more than one lecture at the same time
- Lecturer cannot teach more than one course at the same time
- No room can occupy more than one lecture at the same time
- Some classes require classes to have particular equipment. For example, audio visual equipment,projectors etc

- The capacity of classrooms should match with student size

Soft constraints can be defined as the constraints which can violate them in scheduling, the output is still valid.

- As much as possible, minimize the use of early morning and late evening hours.
- Minimize continuous lectures/blocks of the same course in a day. It is preferred to spread them over the week as much as possible.
- and student should not have an only lecture in any given day

During the last 50 years there were hundreds of researches done about generating time tables and many products were developed as tabulated in Table1.1

Table 1.1 Researches done and Existing products

Researches	Products
"Development of a University Lecture Timetable using Modified Genetic Algorithms Approach" done by Modupe, Alade O., Olusayo, Omidiora E., Olatunde and Olabiyisi S	FET Timetabling Software
"Timetable generation system" done by Chowdhary, Anuja and Kakde, Priyanka and Dhoke, Shruti and Ingle, Sonali and Rushiya, Rupal and Gawande, Dinesh	Time Table Management System developed by Muhammad Jishan for Aligrah Muslim University
"Automated Timetable Generator" done by Sayed, Er Shabina, Ahmed, Ansari, Aamir, Ansari, Zaeem and Ansari	Automated time table generation system developed for Thapar University

When the number of constraints change, developed methods also need to be changed. But these researches and products only answer to an unique set of constraints as following table.

Table 1.2 Considered and Non considered Constraints for Researches and Products

Product/Reaserch	Considered	Non Considered
FET Free Timetabling Software	time constraints,space constraints	Limited some constraint such as Maximum number of working days per week,number of teachers,number of sets of students,number of subjects,number of rooms etc..
Development of a University Lecture Timetable using Modified Genetic Algorithms Approach	1.Didn't violate hard constraints at all. 2.Only two of the soft constraints were violated	some soft constraints not violated such as student's population exceeded the required capacity of the lecture room, etc..
Automated Timetable Generator	1.No students can attend more than one lecture at a time. 2 No lecturer can teach more than one subject at a time. 3 No room can occupy more than one lecture at a time. 4.The lectures are not allotted to time slots which come under the lecturer's.	Didn't consider all constraints both hard and soft

1.2.2 The problem

For the existing method, an expertise with a wide knowledge is required. As this is a constraint satisfaction problem it needs more effort as well as consumes more time. When the number of constraints are increased the complexity and conflicts also increased. And this is repeated each semester and each year consistently. Even after the process, we have experienced changes of the time table at the very beginning of the semester. If there's a change required adjusting existing time table is difficult.

1.2.3 The proposed solution

Main component of our project is the alogorithm that we are going to develop. It'll cost most of the time as we need to figure out what were the problems occured and we had to get rid of those problems as we process. We need inputs like student list, lecturer list, class room list, time slots, course list. Soft constraints such as Lecturer's requirements about time slots and lecture halls will also be inputs to the system. Most of the hard constraints that we discussed earlier will be included in our alogorithm. All the data will be enterd to database.

We are proposing a web based application with three types of accounts as students, lecturers and adminstrations. Initially administration can enter the data or they can connect existing database to our system. Then we let lectures to enter their choices like time, place and so on. For this we will provide a time limit. If someone fails to enter the details system will set default values. Then our algorithm will generate the most appropriate time table and it will be visible to admin and the lecturers. In this time period lectures can make request about changing thier choices. Adminstration can chose whether those request are approvable or not and apply them accordingly. If there is a change, system will generate another time table according to the requirements.

Finally we would show the time table for all the three types of users. Student will see his or her time table for the week and the lecture will see the same. Admin will have all the time table details.

For the web front end we'll use HTML,CSS and angular. Our database will contain many relational features so we'll use MySql. For the web API we would use JAVA and maven spring framework.

If there's anything newly added like talks, industrial sessions organize by the faculty, admin can update it in the time table. Student will able to see whether there is a crashing or not.

We hope to add more features like lectures can cancel the lectures through our system and students will notify. Also we hope to deliver a mobile application for this and it will show the time table for the relevant day.

1.2.4 Deliverable and milestones

Deliverable product:

In our project we hope to deliver a web appication which optimize the timetable generating process with new userfriendly features like giving the corresponding time table to each student and lecturer, ability of giving notifications for cancelling the lectures etc.

As our future work, we hope to implement and deliver a mobile version of this application with specific features for students and lecturers.

Our planed milestones can be specified as follow,

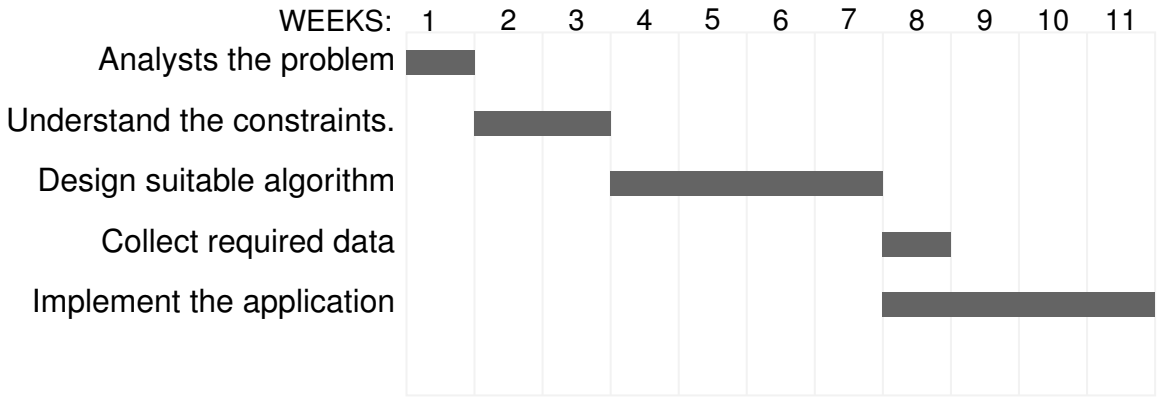

Fig. 1.1 Milestones

Chapter 2

Related works

2.1 Researches

There are many attempts made on generate time tables from automated sytems. Also we can see there are thousands of articles published on the same problem. When we go through these applications and proposals we can see the idea of automating time tables were begin after mid 90's. The research that have been started from Gotlieb in 1963 [1]. can be expressed as the first research done on this category. After that different people used various algorithms to solve the problem. Under those circumstances many time table generators were developed successfully, but with unique set of constraints for each.

In "Proceedings of the 2nd east-west international conference on computer technologies in education" which was published in 1994, Burke, EK Elliman, DG Weare and Rupert introduced a Genetic Algorithm based University Timetabling System [2].In their solution they Provide a choice of several different good schedules from which the user may choose the best and allow database queries to produce a schedule for any staff member, student, room or item of equipment. But their solution directs the timetabler to the most constrained parts of the timetable so that, if necessary, later adjustments should be done manually.

A research about an algorithm to automatically generate schedule for school lectures using a heuristic approach [3] was done in 2012 and published to International Journal of Machine Learning and Computing. Their primary aim was to solve the issue of clashes of lectures and subjects, pertaining to teachers. This solution, however, works from

the teachers' point of view. As well as all the hard constraints (e.g. the availability of teachers, etc.) are resolved rigorously.But their solution consists some limitations as follows

- The number of subjects need to be finalized before the algorithm begins execution.
- Number of teachers entered before execution of the algorithm are assumed to be constant and cannot be changed during or after the algorithm has been executed
- Classrooms for any batch id fixed throughout the day.

Modupe, Alade O and Olusayo, Omidiora E and Olatunde, Olabiyisi S has done a research about the development of a University Lecture Timetable.They have proposed the solution using Modified Genetic Algorithm [4]. In their research they have found that Modified genetic algorithms utilized less CPU time than the Standard genetic algorithm. Further the complexity analysis of both the program volume and programming effort showed that modified genetic algorithms is less in value than standard genetic algorithms. But there was a limitation of violating two of the soft constraints.(student's population exceeded the required capacity of the lecture room, etc..)

In 2014, Chowdhary, Anuja and Kakde, Priyanka and Dhoke, Shruti and Ingle, Sonali and Rushiya, Rupal and Gawande, Dinesh have published their research to International Journal of Computer Science and Mobile Computing about a Timetable Generation System [5]. Easier slot assigning, Less time consumption, NO slot clashes, Always considers the other department slots first, Various possible slot combinations can be acquired are some of the features they have achieved. But have considered unique Hard constraints such as workload and number of lectures per day.

Sayed, Er Shabina and Ahmed, Ansari and Aamir, Ansari and Zaeem, Ansari have researched about an Automated Timetable Generator [6]. In their research they have used the following constraints,

- The teaching hours on an instructor should be limited within the allowed maximum hours.
- A break for lunch/dinner must be allotted to the instructors. 1.No students can attend more than one lecture at a time.

- No lecturer can teach more than one subject at a time.
- No room can occupy more than one lecture at a time.
- The lectures are not allotted to time slots which come under the lecturer's

But very limited hard and soft constraints were considered.

In 2015, Mittal, Dipesh and Doshi, Hiral and Sunasra, Mohammed and Nagoure, R have published their reseach regarding Automatic Timetable Generation using Genetic Algorithm [7]. They had reduced the time required by using Genetic algorithm. In their research they generated a timetable which is more accurate, precise and free of human errors. It was a simplified design which used real data and reduced development time.But it was a unique solution and it was limited in giving professor preferences.

2.2 Existing Products

Some of the applications developed for timetable generation in last 15 years are FET Free Timetabling Software [8] ,Time Table Genaration System by Thapar Univerity [9] and Time Table Management System by Muhammad Jishan [10]. And there are also online tools developed such as The "TimeTable Generator" by Dillon Dixon [11]. Problem regarding this system is we need to each data one by one. That mean we have to create a new database. So it will need more time to enter each detail to the application.

2.2.1 FET Timetabling Software

FET Timetabling Software is a great product build by MarvelSoft. They manage to turn the database input to the time table using this tool. It has limitations such as Maximum number of students, teachers, lecture rooms. Apperantely those limitations seems very good. As an example they describe maximum number of rooms as 6000, maximum nuber of teachers as 6000. But the problem with this product seems to be it binds with "MarvelSoft SchoolAdmin". To use the FET Timetabling Software the school or unversity need to purchase the school management software implemented by them.

2.2.2 Automated timetable generation system used for Thapar University

Thapar University also developed a automated time table generation system for their University. They currently use this system productively. The system use good database design. Also it has a good interactive interface. So it is easy to use and they gurantee error free system. But there are some drawbacks in this system like, requirements can't be changed once finalized, reponse time of the system is not guranteed, unable to generate time table for all the batches of the university once.

2.2.3 Time Table Management System developed for Aligrah Muslim University

Time Table Management System developed by Muhammad Jishan for Aligrah Muslim University is another automated time table which has been developed recently. He has developed a software where one can generate a time table just by uploading database and clicking a button. But this also built only for one university(FCSIT), and it is not integrated with the current database that FCSIT use. He mention with the integration might cause slow reults.

In current situation, our university doesn't use an automated time table generator and it is done by manually. Our idea is to implement an automated time table generator for our university by studying the researches and related applications mentioned above. At the meantime we will try to widen our application which it can be easily used by many universities.

References

[1] K. S. Sandhu, "Automating class schedule generation in the context of a university timetabling information system," 2003.

[2] E. Burke, D. Elliman, and R. Weare, "A genetic algorithm based university timetabling system," in *Proceedings of the 2nd east-west international conference on computer technologies in education*, vol. 1, pp. 35–40, 1994.

[3] A. Nanda, M. P. Pai, and A. Gole, "An algorithm to automatically generate schedule for school lectures using a heuristic approach," *International Journal of Machine Learning and Computing*, vol. 2, no. 4, p. 492, 2012.

[4] A. O. Modupe, O. E. Olusayo, and O. S. Olatunde, "Development of a university lecture timetable using modified genetic algorithms approach," *International Journal*, vol. 4, no. 9, pp. 163–168, 2014.

[5] A. Chowdhary, P. Kakde, S. Dhoke, S. Ingle, R. Rushiya, and D. Gawande, "Timetable generation system," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 2, 2014.

[6] E. S. Sayed, A. Ahmed, A. Aamir, and A. Zaeem, "Automated timetable generator,"

[7] D. Mittal, H. Doshi, M. Sunasra, and R. Nagoure, "Automatic timetable generation using genetic algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 2, pp. 245–248, 2015.

[8] "Fet time table." http://lalescu.ro/liviu/fet/.

[9] "Time table generation system - university of thapar." https://www.scribd.com/doc/7266179/Scope-Statement-for-the-Time-Table-Generation-System-for-Thapar-University.

[10] "Time table management system by muhammad jishan." https://www.slideshare.net/imdzeeshan/time-table-mgt-system.

[11] “Timetable generator by dillon dixon.” https://www.timetablegenerator.io/.