Information Systems International Conference (ISICO), 2 – 4 December 2013

Analysis of Automated Scheduling System Case Study Oral Final Study Examination in Swiss German University

James Purnama, Julius Polar, Maulahikmah Galinium
Department of Information Technology, Faculty of Engineering and Information Technology
Swiss German University, Indonesia

Keywords:

Timetable
Examination timetabling
Heuristic Algorithm
Automated timetabling

ABSTRACT

Timetabling is a complex and often difficult task. Considering the large number of students, availability of lecturers and rooms, scheduling a feasible schedule without mistakes is not an easy task. Some universities are still using the manual paper based scheduling. A computer based timetabling system can greatly help the creation and management of university timetabling. This paper analyzed the problem of Oral Final Study Examination in Swiss German University, and proposed a computer based solution using a heuristic method for an automated scheduling system.

Corresponding Author:

James Purnama,
Department of Information Technology, Faculty of Engineering and Information Technology,
Swiss German University,
EduTown BSD City, Tangerang, Indonesia.
Email: james.purnama[at]sgu.ac.id

1. INTRODUCTION

Timetable is a chart, commonly displayed in table format, which shows one's schedule. Timetabling is a widely studied topic, which includes the schedule of public transportations, task assignments, and self-management. It is one of the most important aspects of time management. Every educational institute has a timetable for lectures and examinations. Creating a timetable is not a simple matter. One has to take into account the amount of students, the number of rooms available for the lecture, and of course, the availability of each lecturer. One also has to make sure that there are no conflicting schedules, such as a lecturer teaching in multiple lectures at the same time. Managing a timetable is sometimes harder than making one. To move a schedule, one has to look for an available room, at a time when the students and lecturers are available, which is sometimes troublesome on a hectic timetable. To make matters worse, timetables are sometimes created and managed manually, without the aid of a computer system, which requires demands resources and is prone to human error.

Some universities have implemented a computer based timetable management systems [1]. These systems are capable of generating a feasible timetable, as well as providing means to manage the timetable. The Swiss German University (SGU) examination office (EXO) is responsible for managing and moderating timetables for examinations, including Oral Final Study Examination (OFSE). The creation of each faculty's OFSE schedule is given to each faculty's administration officer (FAO). Every FAO still uses the manual, human based method of timetabling. A computer based system would help the Examination Office to create a feasible timetable efficiently, manage timetable accurately, and do so with less time and resources.

The purpose of this research is to address the problems with SGU's current system for OFSE scheduling process, as well as analyzing and designing a computer based solution to improve the timetabling of OFSE. This research will test the usability of heuristic based approach of solving the timetabling problem. This research significantly contributes to SGU to improve the overall effectiveness of OFSE scheduling process, to reduce amount of necessary human resources, and to reduce number of mistakes in the process of creating and managing OFSE schedule.

The most difficult part of the timetable process is assigning students to the timetables. One must be careful not to assign a student to two examinations in the same day. This might not be a difficult problem with a low number of students, but as the number of students grows, this becomes a troublesome task. Since humans are not precision machines, errors tend to happen as the number increases. These are some of the common errors during the manual process of timetabling, i.e. a student has multiple exams in one day, a student is assigned to the same subject twice, a student is assigned more than five subjects, a person can be in two different places at the same time.

2. RESEARCH METHOD

The methodology for this research is described as follows:

[23] Literature and related works review: The first step is to gather information related to this research through literature review, in order to know more about, identify the trends of solution for, and form a solution for this research problem. The literature is taken from research journals, research papers, and articles available online on the internet.

[24] Analysis: The second step is to analyze the current system, identify the problems, and gather the requirements for a better system. Information and requirement gathering can be done by interviewing related officers in EXO and FAO. The amount of constraint is also analyzed in order to test the capabilities of heuristic approach.

[25] Design: The third step is designing a system based on the requirements. The proposed system must have the features to solve the problems of manual scheduling.

A computer system can be made in order to help the timetabling creation process by automatically generates a feasible schedule. The system should be able to generate a schedule within a reasonable amount of time, given the amount of parameters and timetabling constraints, as well as eliminates problems raised from human errors [2,3]. Aas with all problem solving, timetabling problems can be solved by generating every single possible schedule within the constraint of available resources. However, this process would take an unrealistic amount of time. A timetable solution usually involves two processes: a heuristic technique to generate a feasible timetable, and a metaheuristic technique to improve the previous solution [3].

The heuristic technique [4] used to generate the timetable is direct heuristics, which simulates a human way of approaching the problem. As a rule, the algorithm will prioritize the most constrained lecturers first, since lecturers are the most contested resources in this problem. This method would be able to solve the timetabling problem and present a feasible timetable, without the need of any sophisticated methods. This is due to the amount of constraint of the timetabling problem itself.

The system would not include a metaheuristic method. It will, however, provide means to modify the resulting schedule. Direct human intervention will result in a schedule which is more acceptable for humans, mostly for aesthetic and convenience reasons. The main requirements for the system are creating a feasible timetable, viewing a timetable and moving and managing schedules. Some other features are viewing timetables based on examiners, students, or subjects, viewing other department schedules, viewing each student's taken subject, viewing each lecturer's subject and availability.

As illustrated in figure 1, the timetabling process starts with FAO inputting the necessary data of students and lecturers. The required data from the system are the student's name, department, and chosen subjects, and each lecturer's name, examined and observed subjects, and their time availability. With all the necessary data inputted, the system can start generating a schedule. The resulting schedule should be confirmed by FAOs for necessary modifications. Afterwards, the schedule is stored in the database for viewing purposes. As illustrated in figure 2, the process is described as follows:

1. The lecturers (both examiners and observers) for each department are listed. That is, lecturers examining or observing a subject for the department.
2. Every subject is listed for each examiner.
3. The examiner's time availability is listed and sorted by the least available.
4. Start looping of examiner's time availability. The loop starts by looping the days, then periods of the day the examiner is available.
5. An observer is found for the examination session. Up until this process, examiner, date, period, and subject are available as searching parameters. An observer is searched for by date, period, and subject parameter. The search rule is: "search for an observer for this subject, who is available for this time, i.e. the observer is not examining or observing a session during that time".
6. A student is found for the examination session. Here, a rule must be followed, that is, OFSE regulations forbid a student to take more than one subject in one day. Therefore, the search rule is: "search for a student is taking this subject, and does not have other exams during this specific time".
7. A room is found for the examination session. The search rule is simple: "search for a room, claimed by this department, which is available during this specific time".
8. This completes all components for an OFSE session. We can now proceed to make an OFSE session

using the values from previous processes. The session is then stored on database. The resulting schedule is used as a constraint for the remaining schedule creation.

9. Continue appending schedules until the end of every loop.

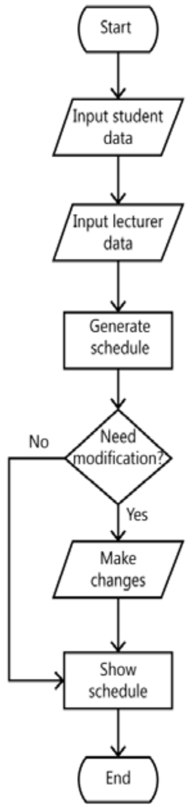


Figure 1. Flowchart of OFSE Schedulling System

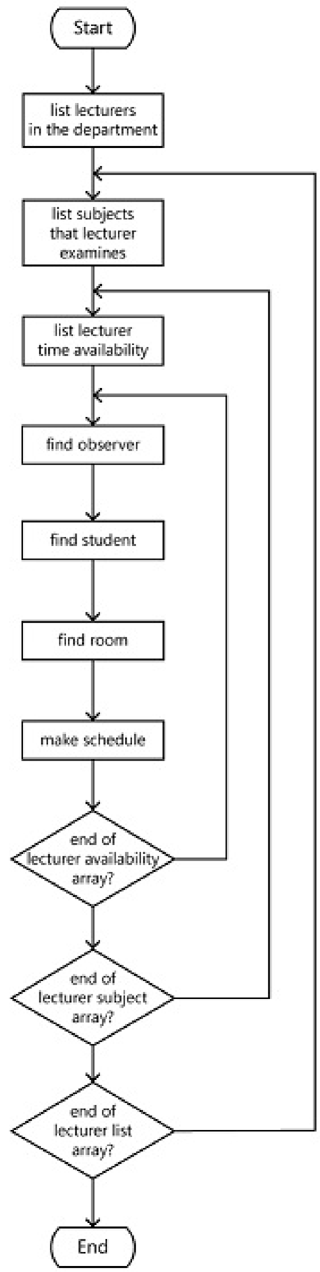


Figure 2. Flowchart of Schedulling Algorithm

The examiner's availability is looped for each different subject a lecturer examines. It is done so to maximize the examiner's time. For example, an examiner is available on Monday, Tuesday and Wednesday, and examines two subjects, A and B. During the scheduling process of subject A, the schedules are made only on Monday and Wednesday because, say, the students may have another exam that day. The following loop for scheduling process of subject B would still go through Tuesday.

The loop order of examiner-subject-availability could be changed to examiner-availability-subject. However, it would tend to create a schedule where an examiner would have to examine multiple subjects in a day. Moreover, it might also create one without unified order, such as period 1-4: subjects ABAB. Examiner-subject-availability also produces multiple subjects in a day, but in an ordered period, such as period1-4: subjects AABB.

The algorithm uses an ascending order when fetching the list of lecturers and students. The least available lecturer is still listed first, then the order of the rest, usually full time lecturers that have the same amount of availability, and can be determined as an input parameter before starting the scheduling process. This makes the algorithm works one way, i.e the algorithm only considers a certain direction during the scheduling. The algorithm can also receives random as the ordering, and it might provide a certain amount of random order during the scheduling process

3. RESULTS AND ANALYSIS

Figure 3 shows the user interface of the schedulling system. When a user opens the application, they have to login to authenticate. Each department will have one account for each FAO. User will be redirected to view a schedule menu after login. The navigation panel on the left shows the available menu. Data input, viewing and schedule creation are available from the navigation panel. From the view schedule menu, the user can choose to view the schedule from multiple categories mentioned previously.

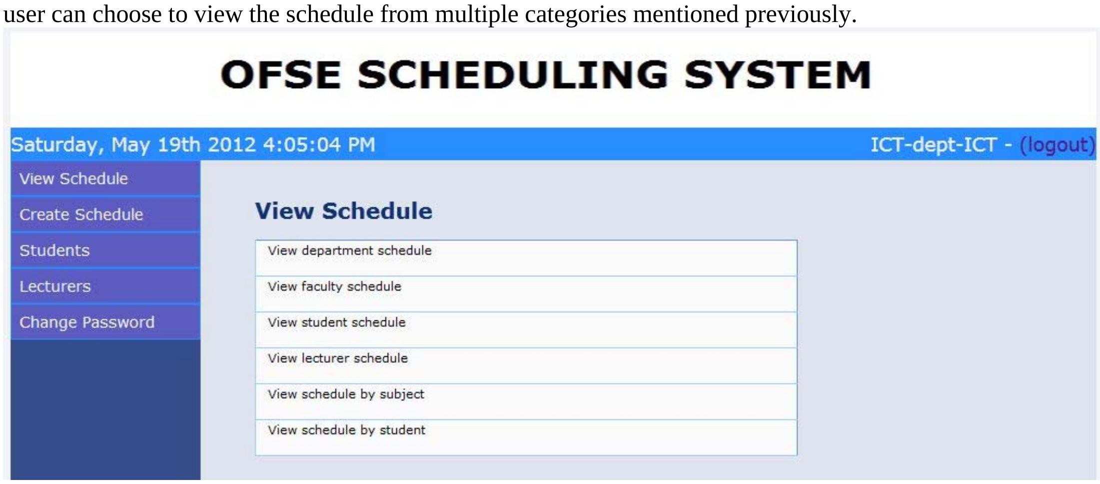
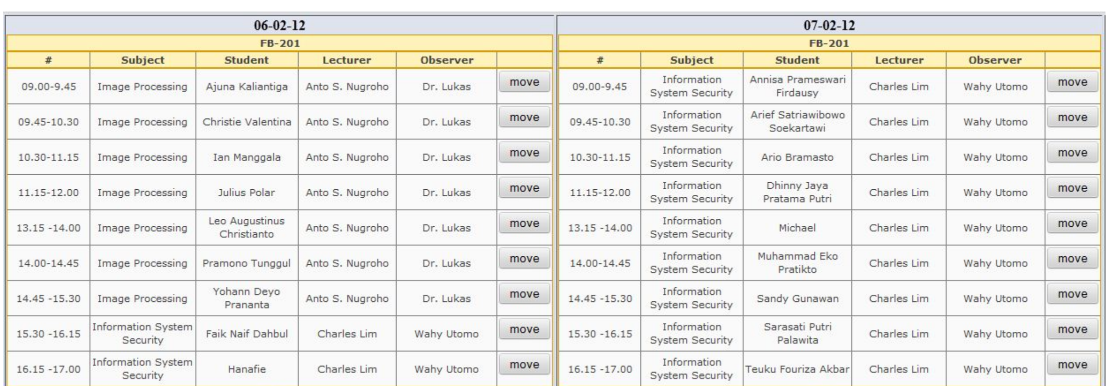


Figure 3. OFSE Scheduling System Main Menu

Figure 4 shows an example of the schedule of the IT department. The schedule will be grouped by days, and by rooms. A session can be immediately moved by clicking the 'move' button.



06-02-12, FB-201

#	Subject	Student	Lecturer	Observer	
09.00-9.45	Image Processing	Ajuna Kaliantiga	Anto S. Nugroho	Dr. Lukas	move
09.45-10.30	Image Processing	Christie Valentina	Anto S. Nugroho	Dr. Lukas	move
10.30-11.15	Image Processing	Ian Manggala	Anto S. Nugroho	Dr. Lukas	move
11.15-12.00	Image Processing	Julius Polar	Anto S. Nugroho	Dr. Lukas	move
13.15 -14.00	Image Processing	Leo Augustinus Christianto	Anto S. Nugroho	Dr. Lukas	move
14.00-14.45	Image Processing	Pramono Tunggul	Anto S. Nugroho	Dr. Lukas	move
14.45 -15.30	Image Processing	Yohann Deyo Prananta	Anto S. Nugroho	Dr. Lukas	move
15.30 -16.15	Information System Security	Faik Naif Dahbul	Charles Lim	Wahy Utomo	move
16.15 -17.00	Information System Security	Hanafie	Charles Lim	Wahy Utomo	move

07-02-12, FB-201

#	Subject	Student	Lecturer	Observer	
09.00-9.45	Information System Security	Annisa Prameswari Firdausy	Charles Lim	Wahy Utomo	move
09.45-10.30	Information System Security	Arief Satriawibowo Soekartawi	Charles Lim	Wahy Utomo	move
10.30-11.15	Information System Security	Ario Bramasto	Charles Lim	Wahy Utomo	move
11.15-12.00	Information System Security	Dhinny Jaya Pratama Putri	Charles Lim	Wahy Utomo	move
13.15 -14.00	Information System Security	Michael	Charles Lim	Wahy Utomo	move
14.00-14.45	Information System Security	Muhammad Eko Pratikto	Charles Lim	Wahy Utomo	move
14.45 -15.30	Information System Security	Sandy Gunawan	Charles Lim	Wahy Utomo	move
15.30 -16.15	Information System Security	Sarasati Putri Palawita	Charles Lim	Wahy Utomo	move
16.15 -17.00	Information System Security	Teuku Fouriza Akbar	Charles Lim	Wahy Utomo	move

Figure 4. Example of Timetable Result

The testing is done by generating a schedule according to several different scenarios. Every scenario has a same number of students, unless mentioned, with the same selection of subjects. Every scenario also uses the same lecturers with different time availability, and they examine and observe the same subjects [5].

The testing process is done as follows:

4. Students' data are inputted into the system, including every student's name and their chosen subject.
5. Lecturers' data are inputted into the system, including every lecturer's name, examined and observed

subject, and their time availability.

Every scenario uses data gained from the SGU IT Department OFSE year 2012. The experiments will not use data from other faculty for some reasons:

1) Every faculty claims their own rooms, so every faculty uses different room; hence every faculty's schedule will not affect another, and can be done separately.
2) IT faculty has plenty of good samples for experiment. Some lecturers are so scarce in time. Some subjects are only taken by one student only. The mandatory subject is even examined by a guest examiner which was not available all the time.

Table 1. Results of Timetabling with Different Scenarios

No	Case Scenario	Result and Comments
1	Lecturers are available everyday on every period.	OFSE is finished in 6 days. Schedules are packed in the first week. Every room is utilized efficiently. Subjects with a big number of students tend to be grouped in a day in a classroom.
2	Lecturers are available according to their time availability in the latest OFSE.	OFSE is finished in 7 days, over a span of 3 weeks. The last 2 weeks only have 1 class due to a unique availability time of a certain examiner.
3	Lecturers have varied and spread time availability. Some full time lecturers' availability is reduced.	OFSE is finished in 10 days, over a span of 3 weeks. Most subjects are spread throughout the weeks. Schedules of a same subject still tend to be grouped in a day.
4	The number of students is doubled.	OFSE is finished in 14 days. The resulting schedule looks similar to case 2. Due to the number of students, the schedules are even more grouped by subjects in a day.
5	Lecturers are very limited in time, with various time availability.	The algorithm could not produce a feasible schedule. The resulting schedule is incomplete.

Table 1 shows the result of each scenario test result. The first case is the best case scenario where every lecturer is always available. The algorithm can utilize the rooms efficiently. The second case is the simulation of latest OFSE. Compared to a human made schedule, the algorithm can assign multiple subjects in one class better than human. The total day required is also less than a human made schedule. A human made schedule requires 10 days, while the algorithm only requires 7 days. The third case is a variation of the second case. In this case, every lecturer has varied time availability spread throughout the three weeks. Full time lecturers' time are reduced and spread. The algorithm successfully produces a feasible timetable. The fourth case is a simulation of faculties with more students. In this case, the number of students is doubled. Some lecturer's time is modified to accommodate the students, in order to maintain a feasible timetabling problem. The algorithm still manages to generate a timetable. The fifth case is the worst case scenario. Every lecturers' time is barely enough to accommodate the number of students. The algorithm could not produce a feasible timetable, but instead an incomplete timetable. This is due to the limited time available, and also, some lecturers uses their time as both examiner and observer. The timetabling process took approximately 1 second for every student. With around 50 students, the scheduling took about 1 minute to complete.

The major weak point of the algorithm is that it is a one way algorithm. It always go in one direction, the least available lecturers. This one way of solving a problem might turn into a dead end. For example, the constraints are such that a feasible schedule can only be generated by scheduling the more available lecturers instead of the least available. This problem can be solved by giving a different order parameter before the initiation. The algorithm can receive one out of three different ordering, which is either ascending, descending, or random, that determines the listing order of lecturers and students.

4. CONCLUSION

Based on several experiments, the system can generate a usable schedule for SGU's OFSE. The primary requirements developed and running well includes data input, schedule viewing, and schedule creation. Even though the system uses a simple direct heuristic method which prioritize lecturer time, the algorithm can solve the timetabling problem, and provide a feasible schedule in many experiment cases. Thus, in this case study, the heuristic algorithm is able to solve the schedule problem of SGU’s OFSE..

5. FUTURE WORKS

In future works, we could also conduct a metaheuristic algorithm to solve the problem. A metaheuristic method could be added on top of the current heuristic algorithm. After the first generation of timetable, a metaheuristic algorithm could improve the resulting timetable to provide a more efficient or acceptable timetable.

ACKNOWLEDGEMENTS

We would like to thanks to the Swiss German University Examination Officer (ExO) for explaining and assisting us regarding the Oral Final Study Examination (OFSE) process in SGU. Their inputs regarding this research are highly appreciated.

REFERENCES

[1] E. Burke, D. Elliman, and R. Weare, "A Genetic Algorithm Based University Timetabling System", *East-West Conference on Computer Technologies in Education, Crimea,* Ukraine, 1994.
[2] E.K Burke and S. Petrovic, "Recent Research Directions in Automated Timetabling", *European Journal of Operational Research, 2002.*
[3] E. Burke, J. Kingston, K.. Jackson, and R. Weare, "Automated University Timetabling: The State of the Art", *The Computer Journal 40 (9) 565-571, 1997*.
[4] H. Arntzen and A. Løkketangen, "A local search heuristic for a university timetabling problem," *the 2002 International Timetabling Competition (TTC 2002),* 2003.
[5] A. Schaerf, "A Survey of Automated Timetabling", *Artificial Intelligence Review 13 (2) 87*-127, 1999.

BIBLIOGRAPHY OF AUTHORS

	James Purnama, S.Kom, M.Kom james.purnama@sgu.ac.id Education: Bachelor Degree: Budi Luhur University, Tangerang, Indonesia Master Degree: Swiss German University, Tangerang, Indonesia Occupation: Lecturer of Information Technology Department and Head of SPQA Department at SGU
	Julius Polar, S.Kom julius.polar@student.sgu.ac.id Education: Bachelor Degree: Swiss German University, Tangerang, Indonesia Master Degree: TU Munich, Germany (in process) Occupation: IT Professional in Uniscon, Germany
	Dr. Maulahikmah Galinium, S.Kom, M.Sc maulahikmah.galinium@sgu.ac.id Education: Bachelor Degree: Swiss German University, Tangerang, Indonesia Master Degree: Lunds Universitet, Lund, Sweden Doctoral Degree: University of Rome Tor Vergata, Rome, Italy Occupation: Lecturer and Deputy Head of Information Technology Department at SGU