Real-Life Faculty Examination Timetabling to Utilise Room Used

San-Nah Sze, Min-Hui Phang and Kang-Leng Chiew

Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak

snsze@unimas.my

***Abstract*—Examination timetabling is an important and yet tedious task to do in every semester. The large number of courses and students increase the difficulty of developing a good examination timetable. Furthermore, the examination timeslots and rooms are very limited in this case study. Therefore, an improved version of two-stage heuristic is proposed and developed a web-based prototype (Faculty Examination Scheduling System, FESS 2.0) to solve faculty examination timetabling problem at Universiti Malaysia Sarawak (UNIMAS). The prototype has been practically used starting from Semester II, 2016/2017. The main objective of the proposed solution is to maximise the room utilisation and minimise the number of rooms for a splitting examination. The outcome of research not only outperform the previous prototype FESS 1.0 but also enhance the services given by faculty management.**

***Index Terms*—Examination Timetabling; Two-Stage Heuristic; Room Utilisation.**

I. INTRODUCTION

Examination timetabling is assigning a set of examinations into a set of timeslots and rooms with the aim of satisfying a set of constraints [1]. It is an NP-hard problem where it required amount of computation to solve the complexity of the problem [2]. The variables such as course, student, timeslot and room may increase the difficulty of scheduling examination timetable.

Over the last ten years, a variety of methods have been applied successfully on solving the examination timetabling problem. The methods are included sequential method [3], parallel metaheuristic [4], genetic algorithm [5], hill climbing search [1] and hybrid hyper-heuristic [6]. The survey by Burke et al. [7], Rankhambe and Pandharpatte [8] had been done for the examination timetabling which are solving by heuristic methods.

II. PROBLEM STATEMENT

This case study focuses on solving real-life faculty examination timetabling. In UNIMAS, each faculty is given the responsibility to plan and schedule faculty's courses for a period of 2 examination weeks. Each examination day has 3 examination slots: morning, afternoon and evening. However, evening examination slot is always the least preferable slot. Besides that, each faculty has a number of faculty owned rooms and some limited usage periods for big shared rooms, which managed by Undergraduates Studies Division (BPPs). Due to the drastic increment in number of students from year 2011 to 2015, which is more than 100%, the objective of the examination timetable solution is to maximise the room utilisation.

In Faculty of Computer Science and Information Technology (FCSIT), the examination timetable was previously scheduled by two experienced planners manually. Due to the size and complexity of the problem nature, it has been unrealistic to solve it manually even just for a feasible timetable. Therefore, Faculty Examination Scheduling System 1.0 (FESS 1.0) which developed by Phang and Sze [9] was introduced and implement to produce a clash-free examination timetable since Semester II 2014/2015. FESS 1.0 is proved not only capable to generate a clash-free examination timetable but also shorten the examination days compared to manually done timetable. However, room utilsation is not considered in FESS 1.0. Some of the examinations were even split into 9 venues which is impractical in real life.

A. Problem Formulation

In this faculty examination timetabling problem, the following notation is used in mathematical modelling.

Let n = Number of courses

Thus,
C = Set of courses: $\{C_1, C_2, \dots, C_n\}$
S_{C_i} = Set of students for course C_i
R = Set of rooms: $\{r_1, r_2, \dots, r_m\}$

There are two types of constraint involved in faculty examination timetabling problem. Hard constraints (HC) are those satisfaction is a must in order to get a feasible solution. Soft constraints (SC) are optional in fulfilling but satisfaction of soft constraints assures better quality solution.

- HC_1 A student can only have one examination at a time.
- HC_2 Room's capacity must fit in the size of allocated exams.
- SC_1 Minimise the total rooms' capacity wastage
- SC_2 Minimise the number of rooms splitting in a course

B. Faculty Examination Scheduling System 1.0 (FESS 1.0)

FESS 1.0 was developed based on a two-stage heuristic as shown in Figure 1.

In Stage I, all the courses were sort based on the course size decreasingly. Then, greedy packing method is used to cluster the courses based on the sorted list. At Stage II, a timeslot that fulfilled the total courses capacity is assigned to each cluster from Stage I. After that, room assignment is done by greedy assignment heuristic, based on the room and course size.

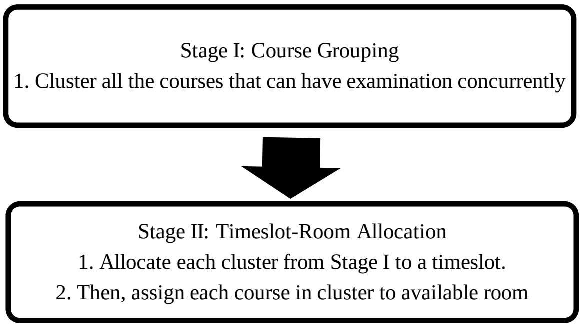

Figure 1: Two-stage heuristic method in FESS 1.0

C. Faculty Examination Scheduling System 2.0 (FESS 2.0)

Since Semester II 2016/2017, improved version of Faculty Examination Scheduling System 2.0 is proposed in order to enhance the room utilisation. As FESS 1.0, the scheduling process in done in two stages.

Stage I: Course Grouping

To reduce the problem scale, course grouping is need where it will group n courses into g groups, where $g < n$.

The algorithm of course grouping has the following steps:

1. For each course i, the pairing ability, P_{C_i} is determined
$$P_{C_i} = \sum_{j=1}^{n} X_{ij} \text{ where } X_{ij} = \begin{cases} 0, & S_{C_i} \cap S_{C_j} \neq \emptyset \\ 1, & S_{C_i} \cap S_{C_j} = \emptyset \end{cases}$$
2. Sort the courses according to P_{C_i}in ascending order. If tie, priority given to the class with bigger number of students
3. For each course i in sorted list, search for an available group $j, j \in g$ which does not has any common student.
4. If the group $j, j \in g$ is available, then add the course into the group, else a new group will be created.
5. Repeat Step 2 and 5 until all the courses are grouped.

Stage II: Timeslot-Room Allocation

After the course grouping and distribution of group in different session, the courses in each group will be allocated into the available timeslot. The algorithm of timeslot-room allocation has the following steps:

1. All the groups that created in Stage I will be sorted based on the total number of student in descending order.
2. Each group will be assigned an examination timeslot which can accommodate all the courses of the group.
3. Sort the courses in each group by the total number of student in descending order.
4. Each course i will search for room, where there be different cases as follow:

Case I: A room fit all the students, but $i \neq n$

- If the partial used room is available, then course i will be set into the room
- Else, the course i will be set into the largest room

Case II: A room fit all the students, but $i = n$

- If the partial used room is available, then course i will be set into the room
- Else, the course i will be set into the smallest room

Case III: Do not fit all the students

- The course i will search for the largest room and the remaining will search for the best fit of room.

Figure 2 shows the example of allocation in a group into the rooms in an examination session. In each group, all the courses will be first sorted based on their total number of student.

- Figure 2 (a): Course TMF2234/TMC1234 fall to Case III which will be split into several rooms.
- Figure 2 (b): Course TMN3053 fall to Case I and will find the largest room.
- Figure 2 (c): Course TMP3414 fall to Case I and will find the room that being used before.
- Figure 2 (d): Course TMP3613 fall to Case II and will find the smallest and fitted room.

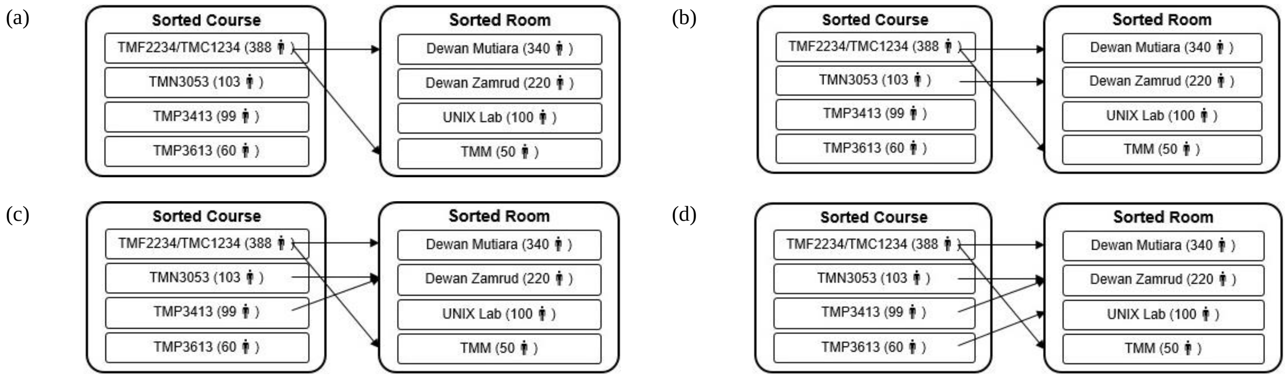

Figure 2: Allocation of courses into available rooms

III. RESULTS AND ANALYSIS

A simulator was created and written in PHP language and using MySQL database to generate the final examination timetable. The simulator is run on 2.30GHz of Intel® Core i5 processor with 4GB RAM.

The real datasets from FCSIT and Faculty of Science Social (FSS) were used for this research.

A. Computational Result

Table 1 shows the comparison between the FESS 1.0 and FESS 2.0. Overall, the computing time of FESS 2.0 is slighter faster than FESS 1.0. For the FCSIT dataset, the total number of exam days was shortened to 6.5 days in FESS 2.0. This is due to the proposed *pairing ability*, P_{C_i} in Stage I, helps in creating less number of group, g. Meanwhile, for the datasets from FSS, the number of exam days maintain 7 days. The main reason behind it is the data nature where FSS has more big common courses across different programs.

Table 1
FESS 1.0 vs. FESS 2.0

Dataset	FCSIT (Sem I, 2016/2017)		FSS (Sem I, 2016/2017)	
No. Of Courses	51		62	
No. Of Students	1593		2038	
Total Enrolment	6335		8163	
Solution	FESS 1.0	FESS 2.0	FESS 1.0	FESS 2.0
Computing Time (sec)	43	30	52	50
Clash Problem	No	No	No	No
Total Exam Days	7	6.5	7	7

B. Room Utilisation

Table 2 shows the room usage during final examination in FCSIT and FSS. The number of rooms used has been reduced regardless the size of the room. In FCSIT, the total number of rooms used was 12 rooms lesser, which is 19% saving by FESS 2.0. Directly, the room capacity left has also improved as much as 72%. Similarly, the total number of room used was decreasing from 86 rooms to 61 rooms in FSS. The room capacity left also been reduced by 57%. This shows that the improved algorithm in FESS 2.0 is effective in improving room utilisation.

C. Rooms splitting in a course

Previously, the number of rooms splitting of a course can be as many as 7 rooms and 9 rooms, at FCSIT and FSS respectively. This solution is impractical and required manual adjustment of rooms assignment after that. Therefore, new venue allocation is presented in FESS 2.0 in order to minimise the rooms used for each course.

Figure 3 presents the improvement result in number of rooms used of a course for both faculties. The maximum number of rooms splitting for a course is reduced to 4, which happen at only a course.

In FESS 1.0, there are more groups created in Stage I compared to FESS 2.0. Furthermore, the range of number of courses in a group is bigger. Some course groups even consist as little as one course in it. This explains the higher number of courses that requiring only one room in FESS 1.0. To overcome this issue, the newly proposed pairing ability P_{C_i} is capable to produce less number of groups, g and reduced the range in number of courses in a group. To improve it further, best fit venue followed by partially used venue are priority in venue assignment in Stage II. These proposed elements not only benefit in room utilisation but also reduced the invigilation duties during exam

Table 2
Room Usage in FCSIT and FSS

	Faculty	FCSIT		FSS	
	Method	FESS 1.0	FESS 2.0	FESS 1.0	FESS 2.0
Size of Room	Small (x < 100)	34	32	48	28
Size of Room	Medium (100 ≤ x < 200)	9	7	7	4
Size of Room	Large (x ≥ 200)	19	11	31	29
	Total Room Used	62	50	86	61
	Improvement (%)	-19%		-29%	
	Room Capacity Used	6335		8163	
	Room Capacity Left	2245	625	2162	922
	Improvement (%)	-72%		-57%	

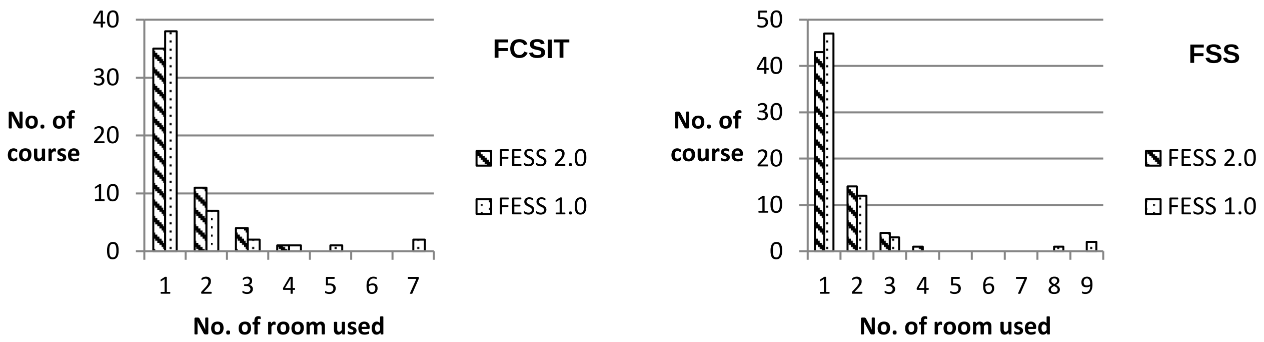

Figure 3: Graph on number of courses against the number of rooms used

IV. CONCLUSION

A two-stage heuristic method was modified and improved to solving the problem of room The improved two-stage heuristic to solve faculty examination timetabling problem is presented. In Stage I, paring ability for each course is assessed in order to cluster the courses more effectively. Meanwhile, more priority rules are proposed in Stage II for venue allocation to achieve minimum number of rooms splitting in a course. Real data from two faculties in UNIMAS is collected for comparison analysis. The computational result shows that room utilisation is significantly improved in both number of rooms used and total unused capacity. Not only that, the proposed solution also minimise the number of rooms splitting of a course and proven the enhancement of the solution practicality. The outcome of this research, FESS 2.0 could also directly improve the manpower utilisation of faculty management in planning and executing exam invigilation. Practically, FESS 2.0 has been implemented since Semester II, 2016/2017 at few faculties in UNIMAS.

ACKNOWLEDGEMENT

This work was supported by UNIMAS Zamalah Graduate Scholarship (ZSU) and a research grant, RACE/1247/2015(03) under Universiti Malaysia Sarawak.

REFERENCES

[1] A. K. Mandal and M. N. M. Kahar, "Combination of graph heuristic with hill climbing search for solving capacitated examination timetabling problem," *2015 4th International Conference on Software Engineering and Computer Systems*, pp. 118 – 123, 2015. IEEE.

[2] R. G. Parker and R. L. Rardin, *Discrete optimization*. Academic Press Professional, Inc., 1988.

[3] E. K. Burke, J. P. Newall, and R. F. Weare, "A simple heuristically guided search for the timetable problem," *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems*, University of La Laguna, 1998.

[4] N. Mansour and G. S. Haidar, "Parallel metaheuristic algorithm for exam timetabling," *Sixth International Conference on Natural Computation*, 2010. IEEE.

[5] M. Hosny and M. Al-Olayan, "A mutation-based genetic algorithm for room and proctor assignment in examination scheduling," *Science and Information Conference*, 2014. IEEE.

[6] R. Qu, N. Pham, R. Bai, and G. Kendall, "Hybridising heuristics within an estimation distribution algorithm for examination timetabling," *Applied Intelligence*, vol. 42, pp. 679-693, 2015. Springer.

[7] E. Burke, D. Elliman, and R. Weare, "Examination timetabling in British universities – a survey," *Practice and Theory of Automated Timetabling*, pp. 76-90, 1995. Springer Berlin Heidelberg.

[8] J. P. Rankhambe and R. Pandharpatte, "A survey on examination scheduling problem (ESP) and hyper-heuristics approaches for solving ESP," *International Conference on Information Processing*, 2015. IEEE.

[9] M. H. Phang and S. N. Sze, *Web-based final examination scheduling system*. (Unpublished degree's thesis), 2015. Universiti Malaysia Sarawak, Malaysia.