How to cite this paper:

Chia-Lih Bong, San-Nah Sze, Noor Alamshah Bolhassan, & Kang-Leng Chiew. (2017). Heuristic faculty course timetabling with student sectioning in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 119-125). Sintok: School of Computing.

HEURISTIC FACULTY COURSE TIMETABLING WITH STUDENT SECTIONING

Chia-Lih Bong[1], San-Nah Sze[2], Noor Alamshah Bolhassan[3], and Kang-Leng Chiew[4]

[1]*Universiti Malaysia Sarawak,* lihlih1206@gmail.com
[2]*Universiti Malaysia Sarawak, snsze@unimas.my*
[3]*Universiti Malaysia Sarawak, bnalamshah@unimas.my*
[4]*Universiti Malaysia Sarawak, klchiew@unimas.my*

ABSTRACT. This paper studies a real faculty course timetabling problem of Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS). FCSIT offers 5 four-year programmes. Each programme has own set of courses but some courses are required by more than one programme. List of courses for each semester of all programmes are planned and suggested by faculty as in course plan but students are free to select their own course registration. FCSIT has to come out with course timetable with no clashes before semester starts. Current curriculum-based timetabling method causes clashes and requires few rounds of adjustment. The aim of this study is to come out with better method than the current curriculum-based timetabling practice. A two-stage heuristic with student sectioning consideration method is proposed to solve the problem by student-based approach. A simulator is developed and tested with real datasets from FCSIT. It generates clash-free timetables in shorter time as adjustment is unnecessary. Results shown that simulator solution performs better in utilise venue resources by reduced 13.1% unnecessary allocation. On top of that, the simulator is proved to be efficient in solving problem for different semesters with changed problem size, different offering courses and different resources given.

Keywords: automated timetabling, two-stage heuristic, student sectioning

INTRODUCTION

Timetabling problem is the problem of assign a number of events into a limited number of resources subject to list of constraints with the aim to satisfy a set of objectives to the highest possible extent [1]. As a well-known NP-complete problem, the degree of difficulty increases enormously with an increasing number of students and courses [2]. The constraints in timetabling can be classified into hard and soft constraints. Hard constraints must be satisfied under any circumstances. Soft constraints are those of their satisfaction is desirable but not mandatory. During the second International Timetabling Competition (ITC) in year 2007, the competition proposed to split university course timetabling problem into two formulations, namely *curriculum-based* [3] and *post-enrolment* (also known as *student-based*) [4]. The constraints and objectives in curriculum-based timetabling are based on the concept of curriculum, which is a set of courses particularly for a group of students (usually grouped by intake and programme). On the other hand, the constraints and objectives are based on the course registrations for student-based timetabling. It considers course registration of each student. Curriculum-based approach has smaller problem size compared to student-based aproach.

Over the last 50 years, a wide variety of methods to tackle timetabling problems have been proposed and tested on real data or benchmark datasets. These methods can be divided into exact, heuristic and metaheuristic. Exact solution is possible only for small problem instances but real-world problems are huge [5]. Later, simple heuristic method is introduced. It produces near-optimal solutions in fast and less complicated manner [6]. The earliest heuristic is graph colouring method. Its main drawback is complex for large scale problem as it would be a challenge to select the vertex to start with. Clustering is another popular heuristic method [7] which aims to reduce problem size and relatively reduce problem complexity [8] [9]. Its main disadvantage is the initial decision on assigning event into groups cannot be changed. In order to improve the solutions' quality, metaheuristic has been introduced. However, it requires high computational expenses as compared to simple heuristic.

Next section discusses the problem statement followed by modelling and formulation. After that, the flow of method used will be explained. The result is shown and the final part concludes this paper.

PROBLEM STATEMENT

Real course timetabling problem at FCSIT, UNIMAS has been used for this study. Current practice for timetabling in FCSIT is curriculum-based. Curriculum-based timetabling fails to consider students who repeat a course and students who enrol courses not according to course plan. This group of students are defined as "repeaters". As result, curriculum-based timetable arise clashes problem. Besides, the number of students' intake has increased tremendously in the last 5 years and predicted to be continued growing in the future. Therefore, insufficient large size venues forces student sectioning. The current practice of student sectioning in FCSIT pre-sets the number of sections to be divided. However, in order to utilise the limited venue resources given, pre-set number of sections is not encourage as it leads to wastage.

MODELLING AND FORMULATION

In FCSIT timetabling problem, the following notation is used in our mathematical modelling.

- Set of courses: $C = \{c_1, c_2, \ldots, c_n\}$, where n = number of courses
- Set of potential students for course $c_i : S_{c_i} = \bigcup_{j=1}^{q} Z_{i_j}$, where q = number of course sections and Z_i = set of course sections for course c_i
- Set of lecture events for course $c_i : E_{c_i} = \bigcup_{j=1}^{b} e_{i_j}$, where b = number of lectures e_{i_b}

The objective function is to minimise the number of unallocated course.

The hard constraints considered for this problem are as follows.

1. A student can only have one lecture at a time.
2. A venue can only be used for one lecture at a time.
3. Venue's capacity must fit in the size of allocated lecture.
4. Total lecture hours of a course must be the number of hour needed for that course.

$$h_i = \sum_{j=1}^{b} h_{e_{ij}} \quad (1)$$

The considered soft constraints are as follows.

1. Lectures of each course should be uniformly distributed over the week.
2. Students' timetable should be uniformly distributed.
3. Maximise venue usage utilisation.
4. Minimise the number of splitting in a course if student sectioning is required.

METHODOLOGY

The proposed heuristic method is divided into two stages. Besides, two types of student sectioning are involve with certain conditions. The algorithm flow chart is shown in Figure 1.

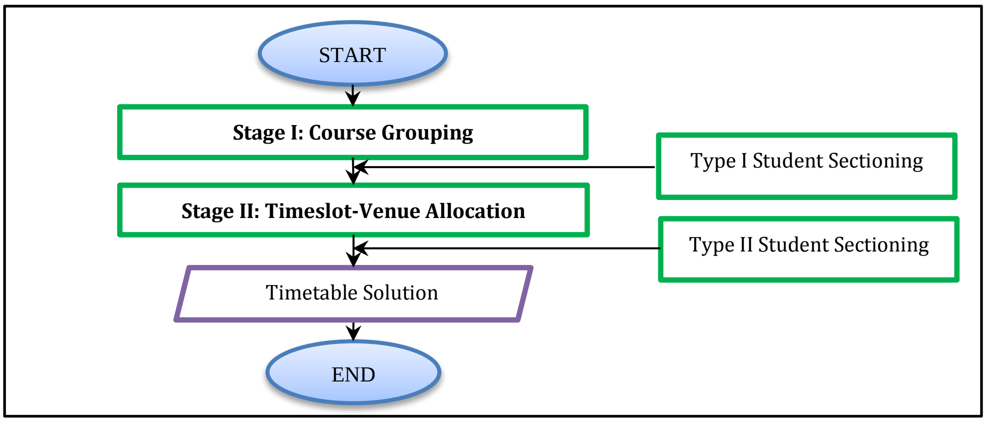

Figure 1. Overall algorithm flow chart.

Stage I: Course Grouping

There are n number of courses, a $n \times n$ matching matrix X is created where courses are listed down the rows c_i and across the columns c_j. The element of x_{ij} is 0 or 1, x_{ij} is 1 if $S_{c_i} \cap S_{c_j} = \emptyset$ (it means c_i and c_j do not have common student) and 0 otherwise. Based on the matching matrix, the pairing ability Y for each course y_i is calculated with Eq. (2).

$$y_i = \sum_{j=1}^{n} x_{ij} \quad (2)$$

Course with higher pairing ability is more flexible to pair with other courses in the same group while lower pairing ability means less flexible. The most difficult course with lowest pairing ability is prior in course grouping. Courses with no common potential student are assigned to the same course group. The highest number of DK and BS available at the same timeslot are identified, in this case 1 DK and 2 BS as in Table 1. The number of DK and BS of a course group are assured not to exceed the available resources.

Stage II: Timeslot-Venue Allocation

List of course groups is generated from Stage I. The task in timeslot-venue allocation stage is assign timeslots for each course group and venues for courses in the course group. Suitable timeslots are identified by matching the timeslot available slot-type with course group demand slot-type based on number of DK and BS available. Balanced timetable is considered by selecting the timeslot of the day with least lectures to allocate. This checking is by looking at the most junior's timetable in the course group as junior enrols most number of courses.

Student Sectioning

Student sectioning is involved in timetabling process when there is unallocated course. Two types of student sectioning are introduced. Type I student sectioning is involved if timeslots available are insufficient to allocate for all course groups. It reduces the number of course group generated and thus reduces the number of timeslots demand. Type II student sectioning is involved if large size venues are insufficient by dividing a large size course into smaller course sections to have events concurrently at different venues. Both types of student sectioning are dynamic splitting which do not pre-set number of course sections. The number of course sections is defined based on the course size and the venue resources available in order to better utilise the venue resources.

RESULT

There are two types of venue resources classified by their management team: university level manages by Undergraduates Studies Division (BPPS) and faculty level in this case FCSIT. Venue resources under BPPS, DK and BS have larger size but limited in number and usage authority. Meanwhile, FCSIT manages venue resources with smaller size, TMM and TR are owned by faculty with full time usage authority. Real datasets from FCSIT has up to 1300 active students and 56 courses with more than 6000 predicted course registrations. A simulator is created written in PHP web based programming language and MySQL database run at 2.60GHz on an Intel® Core i5 processor with 4GB RAM. Table 2 shows the general information and result for three datasets. The computing time of proposed method takes less than one minute to over ten minutes. It varies for different datasets depending on the problem size and requirement of student sectioning.

Table 2. Dataset general information and result.

Dataset	2011/2012-1	2014/2015-1	2014/2015-2
No. of students	520	1279	1236
No. of courses	49	56	56
No. of course registrations	2989	6128	4878
Type I student sectioning	No	Yes	No
Type II student sectioning	No	Yes	Yes
Stage I computing time (seconds)	21	632	29
Stage II computing time (seconds)	7	11	10
Total computing time (seconds)	**28**	**643**	**39**

The simulator solutions are compared with actual solutions as shown in Table 3. Simulator solutions do not have timetable clashes issues while it happens for actual solutions. Despite the effort of student sectioning to avoid unallocated course, there are 4 unallocated courses in simulator solution for dataset 2014/2015-1. However, the actual solutions for datasets 2014/2015-1 and 2014/2015-2 have used extra venues up to 6 BS (initially given 2 BS) and open night time timeslots but simulator solution using initially given resources.

Table 3. Two-stage solutions vs. actual solutions.

Dataset	Actual solution	Simulator solution
2011/2012-1	Feasible solution	Feasible solution
2014/2015-1	Timetable clashes	Clash-free solution (4 unallocated courses)
2014/2015-2		Feasible solution

The main concern in this case study is the student timetable, thus the timetable for each student cluster is analysed. The number of student clusters for dataset 2011/2012-1 is 25 (5 programmes × 5 intakes) but datasets 2014/2015-1 and 2014/2015-2 have 7 intakes with 35 student clusters. A statistic graph on the number of lecture days per week is plotted as shown in Figure 2. 86% of the timetables have 3 or more lecture days per week. More lecture days with less number of lecture hours per day is the aim for a uniformly distributed timetable.

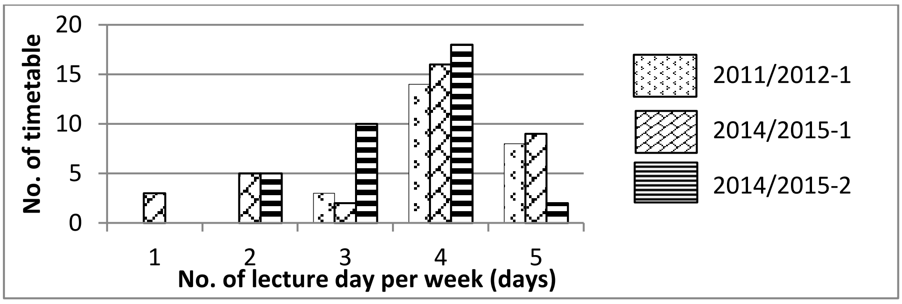

Figure 2. Graph on number of lecture days per week.

On top of that, a statistic graph on the number of continuous lecture hours per day is plotted as shown in Figure 3. The acceptable range is calculated based on the total lecture hours in a week over total number of available timeslots. There are few statistics on more than 5 continuous lecture hours in a day as a result of limited slot-type with more number of large size venues available.

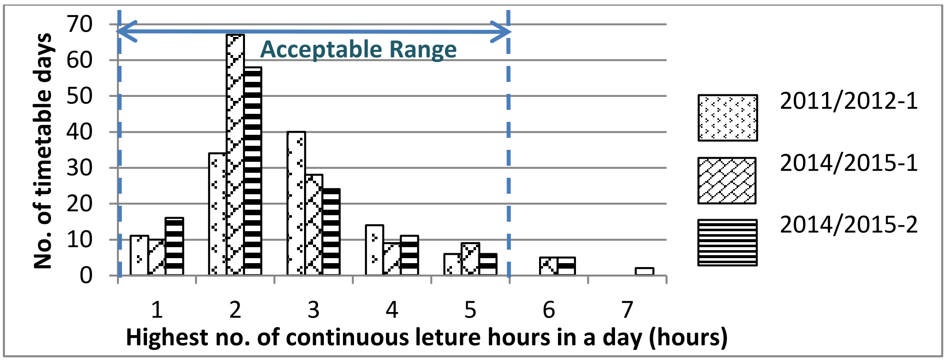

Figure 3. Graph on maximum continuous lecture hours in a day.

Before comparing the venue utilisation, it is noted that real solution used extra resources on top of as given at planning stage. Both methods generate the solution without knowing the actual course registration, thus the size of each course is set by rough prediction during timetabling. In Table 4, the predicted size is classified by venues' maximum capacity and the figures showing the total number of timeslots needed for allocation. Based on total number of venue demand predicted for both solutions, two-stage heuristic has reduced the number by 13.1% for as predicted by real solution. This has already saved up a lot of unnecessary venue resources wastage especially on limited big venues available. It is proven that the real solution has violated the hard constraint, the venue capacity must fit the allocated course. The predicted DK demand was 33 timeslots but only 18 timeslots being allocated. The other 15 DK timeslots demand has been allocated with BS venue which violated the capacity demand of DK. Ideally, the predicted demand and the allocated demand should be matched 100%, which the differ percentage is 0%. However, there do exist differ due to unavoidable constraints on other considerations. By comparing the differ percentage in both solutions, simulator solution allocate better utilisation on venue resources with smaller figures. It even manages to achieve 0% differ for DK demand.

Table 4. Venue allocation comparison for dataset 2014/2015-1.

Venue Demand	Timeslot Available	Real Solution		Differ		Simulator Solution		Differ	
		Predict	Allocated			Predict	Allocated		
DK	22	33	18	-15	45.5%	18	18	0	0%
BS	68	12	67	+55	458.3%	21	42	+21	100%
TMM	48	47	37	-10	21.3%	53	36	-17	32.1%
TR	480	130	100	-30	23.1%	101	97	-4	4.0%
TOTAL Demand		**222 timeslots**				**193 timeslots** (reduced 13.1%)			

CONCLUSION

A two-stage heuristic method with student sectioning algorithm is proposed and developed to solve real case faculty course timetabling problem with student-based formulation. In two-stage heuristic, course grouping stage reduces the problem size by dividing courses into smaller number of course groups. Second stage allocates timeslot for each course group and venues for courses in the course group. Simulator solutions are compared with the actual solutions. Besides, proposed method does not require user to pre-set number of course sections for splitting. Student sectioning algorithm split the course into the most suitable number of course sections based on consideration of its demand, available resources and the course's level of constrained. Therefore, it greatly avoid unnecessary wastage while ensure sufficient capacity for allocated course. This work is customised for FCSIT, UNIMAS. In the future, this work can be improved by increase its generality and flexibility for other institutions.

ACKNOWLEDGMENTS

This work was supported by Zamalah Penyelidikan Naib Canselor (ZPNC) scholarship under Universiti Malaysia Sarawak (UNIMAS) and research funding of RACE/b(3)/1247/2015(03) from Ministry of Education Malaysia.

REFERENCES

Babaei, H., Karimpour, J., & Mavizi, S. (2016). Using k-means clustering algorithm for common lecturers timetabling among departments.

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), 266-280.

De Haan, P., Landman, R., Post, G., & Ruizenaar, H. (2007). A case study for timetabling in a Dutch secondary school. *In Practice and Theory of Automated Timetabling VI* (pp. 267-279). Springer Berlin Heidelberg.

Di Gaspero, L., Schaerf, A., & McCollum, B. (2007). The Second International Timetabling Competition (ITC-2007): Curriculum-based Course Timetaling (Track 3). *Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0*, Queen's University, Belfast, United Kingdom.

Fisher, J. G., & Shier, D. R. (1983). A heuristic procedure for large-scale examination scheduling problems. *Congressus Numerantium*, 39, 399-409.

Gora, W., Lach, G., Lübbe, J., Pfeiffer, O., Zorn, E., & Jeschke, S. (2010). Management and Optimal Distribution of Large Student Numbers. *IEEE EDUCON 2010 Conference* (pp. 1891-1896). IEEE.

Lewis, R., Paechter, B., & McCollum, B. (2007). *Post Enrolment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition*. United Kingdom: Cardiff Business School.

MirHassani, S. A., & Habibi, F. (2013). Solution approaches to the course timetabling problem. *Artificial Intelligence Review*, 39(2), 133-149.

Schaerf, A. (1999). A Survey of Automated Timetabling. *Artificial intelligence review*, 13(2), 87-127.