International Journal of Undergraduate Research 3 (1): 115-120, 2022
e-ISSN: 2682-8189

DOI: 10.26666/rmp.ijur.2022.1.17

SOC Postgraduate Course Timetable: A Timetable System that Minimizes Clashes between Courses

Yim Cheng Poh[1], Juliana Wahid[2,*]

[1,2]School of Computing, Universiti Utara Malaysia, 06010 Sintok Kedah, Malaysia
*Corresponding Author: w.juliana@uum.edu.my

Abstract: The course timetabling problem is common for all academic institutions. One of the Master Program Coordinator's tasks in the School of Computing, Universiti Utara Malaysia, is to produce the postgraduate course timetabling for every semester. Currently, this task is handled manually, which is tedious as several other tasks need to be completed, such as managing other administrative tasks, class preparation, class lecture, student supervision, and managing research. This study proposed the SOC Postgraduate course timetable system that has minimum clashes of courses within the same curricula. Waterfall development is used in developing this system. This system can assist the coordinator in producing timetables more efficiently.

Keywords: *Postgraduate, Course Timetable, System Development, Curricula*

Received: 15 May 2022; Revised: 30 May 2022; Accepted: 10 June 2022; Published: 30 June 2022.

1. Introduction

Education timetables can be classified into the course, examination, and school timetabling [1][2]. Timetabling is common in daily life as many people will plan the schedule or timeline for the things they need to do. Timetabling brings out the meaning of scheduling, planning or arranging the events. People used to plan their timetables manually. Few methods such as drawing by hand and illustrating were used. However, when the era of technology came, people changed their way of doing timetabling.

There are many tools now that can facilitate people to arrange their timetables—for example, Microsoft Excel and a smartphone with a built-in memo.

In the School of Computing (SOC), Universiti Utara Malaysia (UUM), the Master Program Coordinator (MPC) is the person in charge of managing the postgraduate (PG) courses timetable. SOC have two master's programs with coursework mode, such as Master of Science in Information and Communication Technology [MSC (ICT)] and Master of Science in Information Technology [MSC(IT)]. The offering of courses in both programs changed every semester. The MPC will need to plan so that the courses offered will not clash with other courses in the same curricula (program). Therefore, the MPC will need to make more effort in planning the schedule. The workload of the MPC had increased as they had other tasks to complete at the same time. Moreover, planning the schedule that needs to be changed semester by semester was tedious. From this point of view, this study proposed the PG course timetabling system that can produce a timetable with minimum clashes between courses.

Corresponding Author: Juliana Wahid, School of Computing, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia, 60194461710

The following section describes the literature review on university timetabling and the approaches used to solve it. First, the methodology used to develop the proposed system is described in Section 3. Then, Section 4 presented the design and development of the proposed system, while Section 5 illustrated the evaluation procedure and the results of the proposed system. Finally, section 6 summarized the conclusion and future works.

2. Literature review

As this study focuses on PG course timetabling, the reviews are mainly on the course university timetabling problem and approaches for solving it. The university course timetabling problem (UCTP) assigns courses (consisting of one or more lectures) to a timeslot on a specific day and a specific room. The aim is to generate a feasible timetable in which each lecture of a course must be scheduled in a particular timeslot and room, and any two lectures cannot be assigned to the same timeslot. These conditions are categorized as hard constraints in timetabling. Hard constraints are limitations that must be fulfilled so that they can produce a feasible timetable. That means that the production timetable must strictly obey these hard constraints.

In the process of assigning the courses, some other conditions, which are soft constraints, would also be considered; for instance, the number of students attending the course for each lecture must be less than or equal to the capacity of the rooms hosting the lectures, the lectures of each course should be spread across a given number of days, etc. Soft constraints are desirable constraints that are not necessarily needed to create a feasible timetable. However, the more the number of soft constraints fulfilled could increase the quality of the produced timetable. Figures 1 and 2 show examples of the hard and soft constraints.

At the end of the process, the overall objective is to satisfy all the hard constraints and minimize the violation of soft constraints.

Hc1	Capacity of classroom is limited
Hc2	A class cannot have the same subject for more than two lecture periods a day.
Hc3	A class cannot have a lecture with more than one lecturer
Hc4	A lecturer can only deliver one lecture at a time
Hc5	Each lecture is exactly one period long
Hc6	Students can only have one lecture at a time
Hc7	Each lecturer must deliver a specified number of lectures per week
Hc8	A classroom can only be used for one lecture at a time
Hc9	Lessons can be blocked, if required
Hc10	Lectures can be pre-scheduled to a specific time
Hc11	Lecturers' unavailability is considered
Hc12	Allocated rooms must be large enough to accommodate the students
Hc13	Only one lecture for a particular course is allowed
Hc14	Double lectures must include two consecutive periods
Hc15	A set of precedence requirements stating that certain events should occur before others
Hc16	We have different time slots of a lecture day that some of them are overlap

Fig. 1. Hard constraints that had been used frequently by researchers

Sc1	Some lecturers require special facilities. (special tools)
Sc2	Students must not have spare periods or a day with a single session
Sc3	Conflicts between optional subjects chosen by students should be avoided
Sc4	Lecturers can specify times when they prefer not to lecture
Sc5	Some lectures should not take place late in the evening
Sc6	Lecturers' timetables should avoid gaps
Sc7	Rooms should be fully occupied whenever possible, but capacity constraints should not be violated
Sc8	The timetables for rooms should be as compact as possible
Sc9	Lectures should be spread uniformly over the whole week
Sc10	An hour lunch break must be scheduled between 12:00 and 14:00
Sc11	The students provide a sorted list of preferred course
Sc12	Teaching load of a faculty member must be observed
Sc13	The number of lessons per day can be limited
Sc14	The number of students having lunch at a given time should be controlled
Sc15	Lecture rooms should be close to the host department
Sc16	Students should have consecutive lectures in the same building (to avoid moving students)
Sc17	Classes should have lectures either in the morning or in the afternoon
Sc18	Some classes may be split into smaller groups (for example seminar, laboratory, etc) the students need to be equally distributed.
Sc19	All sessions of the course should be scheduled in the same room and the same time-slot, but in different days.
Sc20	Some classes are offered jointly with tutorials, or lab sessions, or both of them
Sc21	Some courses have more than one lecturer
Sc22	Some groups of courses must allocate to special time slots

Fig. 2. Soft constraints that had been used frequently by researchers

The work towards solving UCTP consists of many approaches as described as follows:

In mathematical programming approaches [3], the timetabling problem was decomposed into two sub-problems. Then the sub-problems were solved sequentially. The first was the scheduling of classes to periods and the second task was the assignment of classes to classrooms.

In graph colouring approaches [4], the course period in timetabling is represented by using graphs. For example, construct the timetable with minimum clashes to prevent two adjacent vertices from being in the same color by coloring those vertices. Each color is represented as one of the time periods inside a timetable. After that, the largest degree first, largest weighted degree, and color degree timetabling heuristic will be used to construct a timetable with minimum clashes. Finally, the heuristic will be used based on the difficulty in estimating the timetable.

The cluster method [5][6][7] is a type of method that separates the task or event into different groups according to the hard constraints. Then the groups will need to arrange accordingly to satisfy the soft constraints, but this method could not produce a high-quality timetable result.

In constraint-based approaches [8][9], a timetable containing a set of hard constraints will need to satisfy several soft constraints. Compared to the cluster method, when assigning constraints becomes infeasible, this approach has a backtracking process that can solve the situation and fulfil all constraints.

Metaheuristic methods consist of many approaches such as simulated annealing [10], tabu search [11], genetic algorithms [12], harmony search [13], and hybridization of metaheuristic approaches [14][15][16]. The pros and cons of the method were that the solution produced by this method is of quality, but it requires experiences and people with a foundation in metaheuristic knowledge to use it.

In case-based reasoning [17], the previous timetabling and their reasonable solutions are modelled as a case base. Later, it

will be compared with the new case or the problem to be solved by using a similarity measure or formula.

3. Methodology

The waterfall development methodology was used in developing the PG course timetabling system. Waterfall development consists of several phases: planning, analysis, design, implementation, testing, integration, and maintenance [18].

In the planning phase, literature reviews from SOC websites and SOC PG academic handbooks were conducted to identify current entity features in PG SOC course timetabling. Table 1 shows the entities involved in the PG course timetable, such as the number of courses offered, number of rooms, and number of curricula in the PG SOC program.

TABLE 1. PG SOC PROGRAM FEATURES

Number of courses	Number of rooms	Number of curricula
8	4	2

After the conceptual ideas and essential information had been defined, a requirements analysis was conducted. Requirements of the system had been gathered using the interviewing method with the user of the system, such as MPC and master students, to collect functional and non-functional requirements of the system, which is to help the master student to preside timetable each semester that fulfil with the preferences. The requirements analysis had been documented in the requirements specification, as shown in Table 2. This requirements specification will be used as a reference for the design phase.

In the design phase, many diagrams were formed to show the design and give direction toward the implementation phase. After the requirements were collected and transformed into functional and non-functional requirements, they were analyzed and modelled into charts or diagrams to present their relationship. The requirements were visualized through Unified Modelling Language (UML) into three diagrams which are the use case diagram (Figure 3), class diagram (Figure 4) and sequence diagram (Figure 5). The use case diagram of the system showed what can the SOC Postgraduate Subject Timetable did graphically. It also specified the function of the system. The class diagram is generated to model the elements and used to facilitate building the system. It also acts as the blueprint for the other diagrams. The sequence diagram showed the SOC Postgraduate Subject Timetable flow when the user interacted with the system. It also showed the operation of the system while performing the function. In addition, the design of the user interfaces is also modelled. For example, the interface to key in the student details, the subject details, location of output in the system, and system navigation such as starting and stopping the process. Figures 6 and 7 show the example interfaces of the system prototype.

TABLE 2. REQUIREMENTS LIST FOR DEVELOPING THE SYSTEM

Num	Requirement ID	Requirements Description	Priority
1	REG	Register	
	REG_1	The system allowed the master's program coordinator to register	
	REG_2	The system allowed the master's program coordinator to enter a username and password	
2	LOG	Log in	
	LOG_1	The system allowed the master's program coordinator to login in to the system	M
	LOG_2	The system allowed the master's program coordinator to enter username and password	M
	LOG_3	The system login to the homepage.	M
	LOG_4	The system allowed the master's program coordinator to log out.	M
3	CRT	Create timetable result	
	CRT_1	The system enabled the master's program coordinator to perform timetable scheduling.	M
	CRT_2	The system enabled the master's program coordinator to select the entered data that needed to perform timetable scheduling.	M
	CRT_3	The system enabled the master's program coordinator to preview the timetable result after being processed.	M
4	OUT	Log out	
	OUT_1	The system allowed the master's program coordinator to logout from the system	M
5	USA	Usability	
	USA_1	The system had a nice and interactive interface design.	D
	USA_3	The system allowed the master's program coordinator login successfully with known predefined username and password.	M
6	SEC	Security	
	SEC_1	The system can only log in with a predefined username and password.	M

For the implementation phase, the code of the system had been written. The programming language used was java swing. phpMyAdmin local MySQL database had also been used in developing the system.

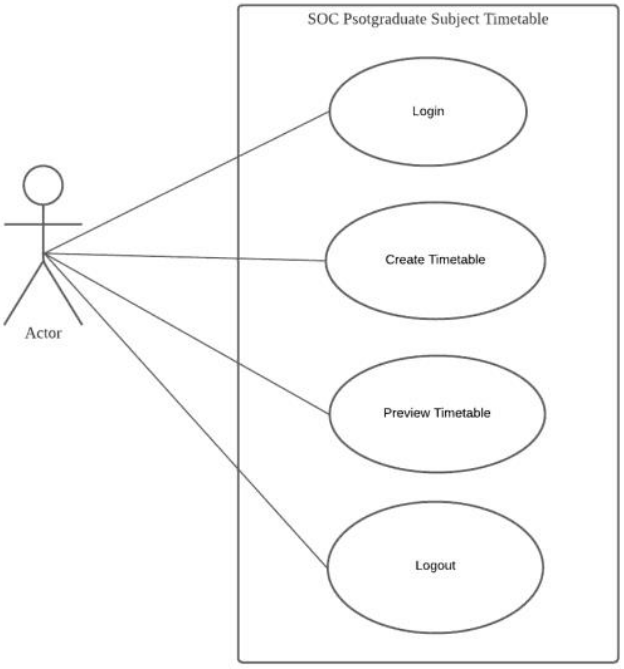

Fig. 3 . Use case diagram for the system

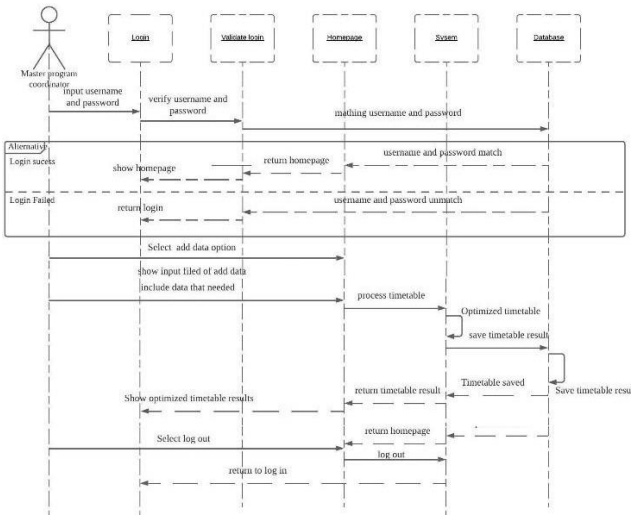

Fig. 4 . Sequence diagram for the system

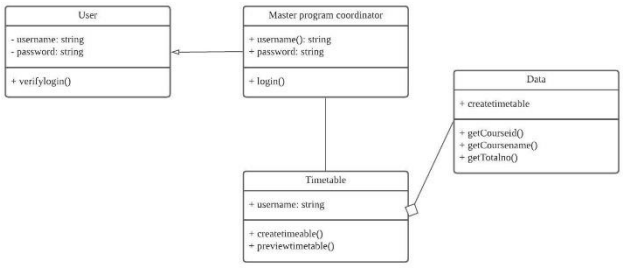

Fig. 5 . Class diagram for the system

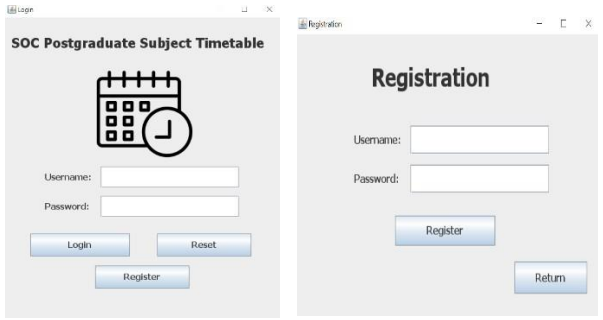

Fig. 6. Interfaces of login and registration

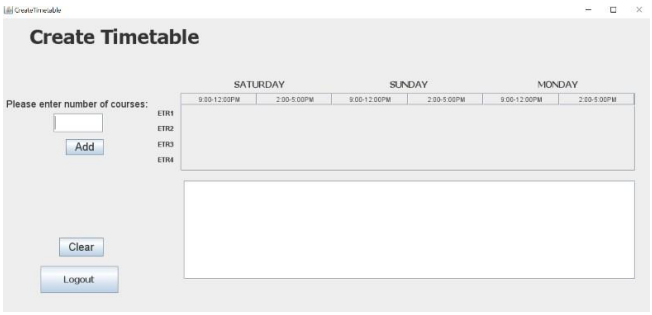

Fig. 7. Interface of the system homepage

In the testing phase, the SOC PG course timetable was evaluated using a moderated usability test method that involves one-on-one sessions between the moderator and participants. Thirty-one respondents participated in this evaluation. The following procedure was implemented in the evaluation process:

a) The facilitator invited the participant by sharing the link to join the Cisco Webex meeting, and the Google Form link was ready. The system was also opened and stood by.
b) The participants joined in the Cisco Webex meeting that the facilitator provided.
c) The participant agreed to the consent form in Google Form.
d) The participant took control of the facilitator's device to interact with the system.
e) The participants run the task that was provided by the facilitator, e.g., login to the system, create a timetable, etc.
f) After completing all the tasks, the facilitator controlled the device.
g) Lastly, the participant must answer the post-task questionnaire through the Google Form (continue from the agreed consent form Google Form).

4. Result of the evaluation

This section analyzed the responses from the participants. As shown in Figure 8, the respondents were primarily female, with 58.1% weightage out of 100% and 16 out of 31 respondents. The respondents were mainly in the age group of 21-25 years old. Almost half of the respondents will plan their schedule or timeline to arrange their to-do list. In respondents who responded yes to their habit of planning their schedule, tools frequently used in

making their schedule or timeline by the respondents were Microsoft Excel. The evaluation of the timetabling system usability consisted of 17 statements. The responses have been tabulated in Table 3.

From Table 3, it can be seen that the overall usability of the SOC PG course timetable has more than average satisfaction. The first statement indicates that respondents are mostly satisfied with the system. In terms of the usability of the timetabling system, it is also more than average agreed that the system performed well. Statements 8 to 13 represented the ease of use of the system. The response showed that the system is easy to use. However, three respondents strongly disagree with statement 11, indicating that written instruction is important in helping new users interact with the system. Finally, the satisfaction of the timetabling system showed that most respondent responses agree and strongly agree, partially neutral and minor responses disagree. It indicates that although the system usability is more than average, enhancements are needed.

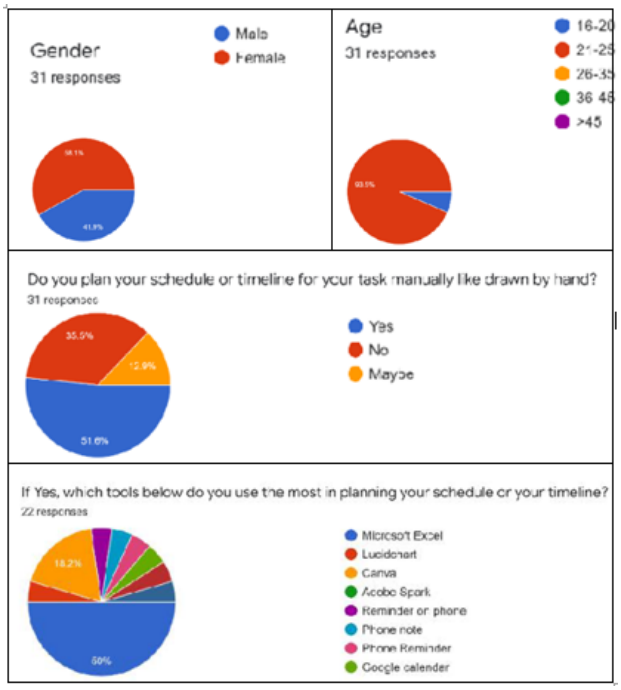

Fig. 8. Demography and background information of the respondents

TABLE 3.USABILITY EVALUATION OF SOC PG COURSE TIMETABLE

Statement	Strongly Disagree (%)	Disagree (%)	Neutral (%)	Agree (%)	Strongly Agree (%)
1	0	0	16.1	41.9	41.9
Usability of the timetabling system					
2	0	0	19.4	45.2	35.5
3	0	3.2	6.5	38.7	51.6
4	0	0	12.9	45.2	41.9
5	0	3.2	12.9	38.7	45.2
6	0	3.2	9.7	41.9	45.2
7	0	0	6.5	45.2	48.4
Ease of use of the timetabling system					
8	0	3.2	16.1	29	51.6
9	0	9.7	6.5	35.5	48.4
10	0	0	19.4	29	51.6
11	9.7	12.9	12.9	25.8	38.7
12	0	6.5	19.4	22.6	51.6
13	0	9.7	9.7	29	51.6
Satisfaction with the timetabling system					
14	0	0	12.9	38.7	48.4
15	0	3.2	6.5	32.3	58.1
16	0	6.5	6.5	32.3	54.8
17	0	3.2	22.6	35.5	38.7

5. Conclusion

Efficient and effective timetable tasks are essential for academic institutions, so that teaching and learning can happen using optimized resources. It is also part of a smart campus initiative at UUM. As many postgraduate students are doing their master's study part-time, timetabling is necessary to allow them to enrol in the courses conveniently that do not overlap with their working time. Therefore, this paper described the development and evaluation of the SOC PG course timetable, a timetabling system that can minimize clashes between courses in the same curricula. After the evaluation process had been done, it could be noticed that improvement was needed. Moreover, a system engine that uses an algorithm will also be a part of the plan to improve further or develop the system so that the functionality and efficiency of the system could be more adaptive to the problem. The burden of the program coordinator would also be expected to be decreased after using the system.

6. Acknowledgements

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

7. References

[1] S. Petrovic, E. K. Burke, University Timetabling, 2004.

[2] S. A. MirHassani, F. Habibi, Solution approaches to the course timetabling problem, Artificial Intelligence Review, Vol. 39, No.2, pp. 133-149, 2013.

[3] J.A. Ferland, S. Roy, Timetabling Problem for University as Assignment of Activity to Resources. Computers and Operational Research, Vol. 12, No. 2, 207-218, 1985.

[4] D. De Werra, Some combinatorial models for course scheduling. Practice and Theory of Automated Timetabling, PATAT, Lecture Notes in Computer Science, Vol. 1153 (1995) 296-308.

[5] N. Balakrishnan, A. Lucena, R.T. Wong, Scheduling examinations to reduce second-order conflicts. Computers and Operations Research Vol. 19, 353–361, 1992.

[6] J.G. Fisher, D.R. Shier, A heuristic procedure for large-scale examination scheduling problems. Technical Report 417,

Department of Mathematical Sciences, Clemson University. 1983

[7] G.M. White, P.W. Chan, Towards the construction of optimal examination timetables. INFOR, Vol. 17, 219–229, 1979.

[8] S.C Brailsford, C.N. Potts, B.M Smith, Constraint satisfaction problems: Algorithms and applications. European Journal of Operational Research Vol. 119, 557–581, 1999.

[9] G.M. White, Constrained satisfaction, not so constrained satisfaction and the timetabling problem. In: A Plenary Talk in the Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling, University of Applied Sciences, Konstanz, August 16–18, 2000, pp. 32–47.

[10] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf, T. Urli, A simulated annealing approach to the curriculum-based course timetabling problem. 6th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2013).

[11] Z. Lu, J.K. Hao, Adaptive Tabu Search for course timetabling. European Journal of Operational Research, Vol. 200, No. 1, 235-244, 2010

[12] J.B. Matias, A.C. Fajardo, R.P. Medina, A hybrid genetic algorithm for course scheduling and teaching workload management. 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018, 1–6.

[13] J. Wahid, N.M. Hussin, Harmony Search Algorithm for Curriculum-Based Course Timetabling Problem. International Journal of Soft Computing and Software Engineering, Vol. 3, No. 3, 365-371, 2013.

[14] J. Wahid, N.M. Hussin, Hybrid harmony search with great deluge for UUM CAS curriculum-based course timetabling, J. Telecommun. Electron. Comput. Eng., vol. 9, 33-38, 2017.

[15] J. Wahid, N. M. Hussin, Solving Curriculum-Based Course Timetabling by Hybridizing Local Search Based Method within Harmony Search Algorithm, in M. Berry, et al., "Soft Computing in Data Science SE - 14," Springer Singapore, vol. 545, pp. 141-153, 2015.

[16] J. Wahid, N.M. Hussin, Harmony Annealing Algorithm for Curriculum-Based Course Timetabling Problem. 2012 International Conference on Computer And Information Science (ICCIS2012), 12-14 June 2012.

[17] E. K. Burke, S. Petrovic, R. Qu, Case-based heuristic selection for timetabling problems. Springer Science + Business Media, LLC 2006.

[18] A. Dennis, B. H. Wixom, R. M. Roth, Systems analysis and design, John Wiley & Sons, 2008.