14thAnnual International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), February 12-14, 2024
Publisher: IEOM Society International, USA
Published: February 12, 2024 DOI: 10.46254/AN14.20240448

A Practical Algorithm for Midterm Exam Scheduling in Higher Education Institutions

Hakan Gultekin and Sayyad Zahid Qamar
Mechanical and Industrial Engineering Department
Sultan Qaboos University, Muscat, Oman
hgultekin@squ.edu.om, sayyad@squ.edu.om

Hassan Al-Lawati
Electrical and Computer Engineering Department
Sultan Qaboos University, Muscat, Oman
hlawati@squ.edu.om

Abstract

The efficient scheduling of midterm examinations at higher education institutions poses a multifaceted challenge. Multiple objectives must be optimized, such as minimizing the total number of students who have multiple exams on a single day or on consecutive days, minimizing the number of exams assigned to weekends, etc. The problem also includes many practical constraints belonging to different stakeholders. Some of these constraints are scheduling all exams on the allocated time period, scheduling single section courses to their lecture times, and scheduling all sections of multi-section courses at the same time in the evening or weekends. This study develops an algorithm and a Decision Support System (DSS) for this complex problem. Besides necessary structural constraints, this DSS incorporates real-world complexities such as preferences for exam dates, fixing course dates, and variations in scheduling for single and multi-section courses. A construction heuristic empowered by a local search procedure is proposed, which is implemented using Visual Basic for Applications (VBA). The user-friendly interface facilitates both scheduling and sensitivity analysis, allowing educators to fine-tune parameters like the setting of the number of weeks allocated for midterms and penalties for unfavorable days and time slots. The system's performance is evaluated using real data, demonstrating its potential to enhance educational management.

Keywords
Midterm exam scheduling, Decision support system, Heuristic algorithms, Multi-objective problem

1. Introduction
The major problem in university midterm timetabling is to allocate the midterm exams to dates and timeslots without clashes and overloads. The problem involves many constraints stemming from the university, college, and department regulations, as well as the course requirements. Constraints can be divided according to their structure. Hard constraints are the set of conditions that cannot be violated and must be satisfied. As an example, a student cannot have more than one exam at the same time. Soft constraints are the set of conditions that are allowed to be breached but the target is to keep this to a minimum. They help define the objective function of the mathematical model under a multi-objective setting. For example, a student taking two exams on the same date is not desirable, but allowable. This can be converted to an objective function as minimizing the number of students taking two exams in consecutive time slots.

In many universities, the timetabling process is decentralized; the midterm exam dates are decided by the course instructors. Sometimes, the course instructor decides on the dates by himself and announces it to the students. In some cases, the dates are decided by a mutual agreement between the students and the course instructors at the beginning of the semester. In both cases, due to interdependencies between the courses, several soft constraints will end up with undesirable outcomes, especially when students are taking courses from different colleges (or schools). In some other universities, the timetabling is managed by the department or by the college in a centralized way. The midterm exam dates of all courses are announced by the department at the beginning of the semester. However, scheduling tens or hundreds of courses at the same time by considering all the hard and soft constraints is a complex task and, in many cases, this is done manually. As a consequence, the scheduling process takes a lot of time and effort, is subject to human error, and the quality of the timetable is not satisfactory. In this study, we consider this problem and develop a computerized algorithm to solve this complex multi-objective problem.

Solving the timetabling problem will have a direct impact on several levels, positively affecting the various stakeholders Such as students, instructors, and timetable designers (administration and faculty). Some of the major benefits will be: (a) clash-free exam timetables; (b) minimization of the number of students who are taking more than one exam on the same day or on consecutive days; and (c) more compact timetables, reducing the unnecessary idle time between exams in a single day, while still ensuring some resting time to get refreshed; (d) the instructors will have better exam timetables with enough time to prepare and grade the exams; and (e) the administrative staff responsible for preparing the timetables will be less burdened (both time and effort) since the decision support system will provide better plans with an easy-to-use interface.

2. Literature Review

Timetabling has found an important place among all scheduling problems. Educational timetabling is an NP-Hard problem faced by many higher education institutions. Ceschia et al. (2023) categorized educational timetabling as University Examination Timetabling, High School Timetabling, and University Course Timetabling. In the same study, they provided a total of 6 mathematical programming formulations and their related data sets, also considering the subcategories of these main categories. Qu et al. (2009) also present a good review of educational timetabling.

Mathematical programming is the fundamental method used to obtain exact solutions in timetabling. Al-Yakoob et al. (2010), Bazari et al. (2023), McCollum et al. (2012) and MirHassani (2006) used mathematical modeling for the problem. MirHassani (2006) and (McCollum et al. (2012) mentioned that timetabling problems include a set of hard and soft constraints depending on the problem restrictions. McCollum et al. (2012) used the following hard constraints in their model: All exams must be allocated.

- If two exams have common students, they cannot be assigned to the same time slot.
- The same exam cannot be split into time slots.
- The same exam cannot be split into rooms.
- The seating capacity of the class cannot be exceeded.
- No extra time can be added to a period.
- All ordering requirements are to be obeyed.
- All class requirements are to be obeyed.

They also mentioned the following soft constraints, where minimization is the objective:Number of students taking more than one exam in consecutive time slots.

- Number of students taking more than one exam in a day.
- The period that all exams are spread through.
- Front load: the most crowded course is assigned first.

(Carlsson et al., 2023) considered the exam timetabling problem in Italian universities. They provided a constraint-pbased programming and two mixed integer programming (MIP) formulations. However, MIP formulations fail to provide solutions for large-sized problems in reasonable run times because they are classified as NP-Hard problems. To overcome this, many heuristic/metaheuristic algorithms were also proposed. (Abou Kasm et al., 2019) developed an MIP formulation for the problem that included restrictions on how many exams can the students take in a given time window. However, to solve real-sized problem instances they developed a modified graph coloring heuristic. (Abdul-Rahman et al., 2014) also used graph coloring heuristic for the exam timetabling at Universiti Malaysia Pahang. (Kahar and Kendall, 2010) also considered the exam timetabling problem at the same university and developed a construction heuristic. Their problem considered the distance between the exam rooms and splitting the exams to multiple rooms.

Other authors (Arbaoui et al. 2016; Zhu et al. 2022; Carlsson et al., 2023; Güler et al. 2018; Güler et al. 2021) proposed different solution methodologies for the exam timetabling problem, such as a matheuristic algorithm, an artificial bee colony heuristic, a simulated annealing metaheuristic, an MIP formulation, and a decomposition algorithm.

In contrast with all of the above published literature, this study considers other practical constraints relevant to our institution such as differences between available slots for single and multi-section courses. An efficient construction heuristic is developed and implemented in MS Excel VBA, which has user-friendly features to handle data processing and solution reporting. Also provided are the results of a case study and insights on the number of spanning days and the objective functions.

3. Problem Definition and Solution Procedures

Cohort in our system refers to the year in which a student started his Bachelors degree. Students from the College of Engineering are required to take some Math and Science courses from the College of Science. Midterm schedule of these courses is announced independently by the College of Science. The timetable considers the following rules and restrictions relevant to our College and University; all of these may not be applicable to other institutions.

- Maximum time allocation for each midterm exam is 2 hours.
- Most of the courses in the curriculum have a single midterm exam. These exams must be distributed over four weeks in a way that helps students to have enough time to prepare for each exam. The exams ``end of withdrawal from the courses. The exam results must be announced before the end of the withdrawal period so that the students can make a decision about withdrawing from a course or not. There are also several courses with two midterm exams. The first of these must be scheduled before week 7 and the second one must be scheduled after week 10. These two types of courses are independent of each other with respect to their midterm timing. Considered here are only the courses with a single midterm exam.
- Some courses have a single section whereas the others have multiple sections. For the multi-section courses, the midterm exams of all sections must be scheduled at the same time. For single-section courses, midterm must be during the lecture time. Class timings for undergraduate courses are 8 am to 6 pm (Sunday to Wednesday) and 8 am to 2 pm on Thursdays. For multi-section courses, midterm timing should be clash-free; so the possible timings are 6-8 pm (Sunday to Wednesday), or after 2 pm on Thursdays, or after 8 am on Saturdays. Note that, in Oman working days are Sunday to Thursday; Friday and Saturday are weekends; and no exams can be scheduled on Fridays.
- The soft constraints (objectives) of the problem in order of their priorities are as follows:
  - No two exams on the same day without a break between them.
  - No two exams on the same day.
  - At least one day off between two exams.

  Note that the soft criteria are tried to be satisfied as much as possible. However, a timetable that does not satisfy a soft constraint is still feasible but can be considered as a non-optimal timetable.

Some years back, there was no coordinated Midterm timetable in our Department. Each faculty would announce Midterm dates of their courses. Hit-and-trial and firefighting type of adjustments would be repeatedly done in case of clashes. The improved current system of midterm timetabling in our Department is a manual system, which is done by the faculty member in-charge of Curriculum and Timetabling (courses and exams). The exam dates must be determined at the beginning of the semester and must be announced to the students before the end of the second week of the semester. It takes approximately three days of intermittent work to get the final timetable, and only one feasible alternative is prepared. It is a traditional way of preparing timetables based on the experience of a person, and the new timetabling in-charge has to be trained in this manual system when he replaces the previous faculty. The necessary information for timetabling is collected from multiple resources:

- In order to assign the single section courses to their lecture times, the course timetable is necessary. This is obtained from the Admissions and Registration department of the university.
- To avoid conflicts between exam times, the degree plans of different cohorts are needed. In our department there are two undergraduate programs, Mechanical Engineering and Industrial Engineering, and a joint program with the Department of Electrical and Computer Engineering (Mechatronics Engineering). The degree plans of all cohorts of these two programs are available with the department curriculum and timetabling coordinator.

The drawbacks in the current system are:

- The degree plan for a certain cohort lists the courses that the students should register in a particular semester. The current midterm plan considers only the degree plans of the student cohorts and tries to satisfy the hard and soft constraints for the courses in the same semester of the same cohort. The cohorts are considered independent of each other. Therefore, it cannot detect if some of the soft constraints are violated for the students who take courses from other departments or colleges or who take courses behind or ahead of their schedules unless the students complain about such conflicts.
- It does not consider the exam timetables of other departments and colleges.
- It does not consider the actual number of students that are common in the courses. Therefore, if a soft constraint is violated, it is done randomly, independent of the number of affected students.
- It needs more time and effort to prepare since it is a hit-and-trial manual system.
- The criteria to evaluate the performance of a timetable are not well designed. Hence, the actual performance of a prepared timetable is not known.

This study develops a solution method for this problem to overcome these drawbacks. In our proposed system, we use the courses timetable as before. However, instead of the cohort degree plans, we use the actual course registration data, which includes the class lists for all courses. This is collected from the Admissions and Registration office. In a regular semester, there are approximately 50 courses within the department and 150 courses within the college that must be scheduled. As a result, the class list contains approximately 1,500 students within the department and 6,500 students within the college.

Once the class lists are obtained, the first task is to process the data and convert it to a matrix that denotes the common number of students between any pair of courses. We developed a VBA code that processes the data and generates this matrix, which is approximately a 50×50 matrix at the department level and 150×150 at the college level. Entry a_{ij} of this matrix denotes the number of common students between courses i and j. Also, we generate another three-dimensional binary matrix, for which entry b_{itd} takes the value of 1 if course i have a class at timeslot t in day d. Note that, for multi section courses, these include not the lecture times but weekday evening times and weekends.

3.1 Mathematical Model

In this section we develop the MIP formulation for the problem. In this study, we consider three objectives with their respective priorities. The objective with the highest priority is minimization of the number of students taking exams in consecutive timeslots. We will denote this objective by CS. The second objective is minimization of the number of students taking more than one exam on the same day in nonconsecutive slots which will be denoted by SD. The last objective is the minimization of the number of students taking exams on consecutive days which will be denoted by CD. The objective function of the MIP model is the weighted sum of these objectives. We use the weights γ, β, and α to prioritize the objectives. Here we assume $\gamma \gg \beta \gg \alpha$. Their values are decided by the decision makers, in our case, the Assistant Dean for Undergraduate Studies (ADUS) at our College.

Sets:

Set of courses: $C = \{1, \dots, c'\}$
Set of timeslots: $T = \{1, \dots, t'\}$
Set of days: $D = \{1, \dots, d'\}$

Parameters:

$$b_{itd} = \begin{cases} 1, \text{ if course } i \in \mathrm{C} \text{ have class at timeslot } t \in \mathrm{T} \text{ in day } d \in \mathrm{D} \\ 0, \text{ otherwise} \end{cases}$$

a_{ij} = number of common students between courses i and j
γ = coefficient represents the importance level for minimizing CS
β = coefficient represents the importance level for minimizing SD
α = coefficient represents the importance level for minimizing CD

Decision Variables:

$$x_{itd} = \begin{cases} 1, \text{ if exam of course } i \in \mathrm{C} \text{ is assigned to timeslot } t \in \mathrm{T} \text{ in day } d \in \mathrm{D} \\ 0, \text{ otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, \text{ if exam of course } i \in \mathrm{C}\ is \text{ assigned to timeslot } t \in \mathrm{T} \text{ in day } d \in \mathrm{D} \text{ and course } j \in \mathrm{C} \text{ is assigned to} \\ \text{timeslot } t+1 \in \mathrm{T} \text{ in day } d \in \mathrm{D} \\ 0, \text{ otherwise} \end{cases}$$

$$w_{ij} = \begin{cases} 1, \text{ if exam of courses } i \in \mathrm{C} \text{ and } j \in \mathrm{C} \text{ are both assigned to day } d \in \mathrm{D} \\ 0, \text{ otherwise} \end{cases}$$

$$u_{ij} = \begin{cases} 1, \text{ if exam of courses } i \in \mathrm{C} \text{ and } j \in \mathrm{C} \text{ are assigned to day } d \in \mathrm{D} \text{ and } d+1 \in \mathrm{D}, \text{ respectively} \\ 0, \text{ otherwise} \end{cases}$$

Using this notation, we can formulate the problem as an MIP as follows:

Model:

$$\text{Min z} = \gamma \sum_{i \in C} \sum_{\substack{j \in C \\ j > i}} a_{ij} y_{ij} + \beta \sum_{i \in C} \sum_{\substack{j \in C \\ j > i}} a_{ij} w_{ij} + \alpha \sum_{i \in C} \sum_{\substack{j \in C \\ j > i}} a_{ij} u_{ij} \tag{1}$$

Subject to:

$$\textstyle\sum_{d \in \mathrm{D}} \sum_{t \in \mathrm{T}} x_{itd} = 1 \qquad \forall\, i \in \mathrm{C} \tag{2}$$

$$x_{itd} \le b_{itd} \qquad \forall\, i \in \mathrm{C}, t \in \mathrm{T},\ d \in \mathrm{D} \tag{3}$$

$$y_{ij} \ge x_{itd} + x_{j(t+1)d} - 1 \qquad \forall\, i,j \in \mathrm{C}, t \in T \backslash \{t\} d \in \mathrm{D} \tag{4}$$

$$w_{ij} \ge \left(\textstyle\sum_{t \in \mathrm{T}} x_{itd} + \sum_{t \in \mathrm{T}} x_{jtd} \right) - 1 \qquad \forall\, i,j \in \mathrm{C}, t \in \mathrm{T},\ d \in \mathrm{D} \tag{5}$$

$$u_{ij} \ge x_{itd} + x_{it(d+1)} - 1 \qquad \forall\, i,j \in \mathrm{C}, t \in \mathrm{T},\ d \in D \backslash \{d\} \tag{6}$$

$$x_{itd}, y_{ij}, u_{ij}, w_{ij}, \in \{0, 1\} \qquad \forall\, i,j,t,d \tag{7}$$

The objective function (1) minimizes the weighted sum of three objectives: CS, SD, and CD. Constraint (2) ensures that all mid-term exams will be scheduled. Constraint (3) forces the system to schedule the exams in specific timeslots. It will schedule the mid-term exams of single section courses in their course times and the midterm exams of multi-section courses in the appropriate off-class times. Constraint (4) identifies the courses that are assigned to consecutive timeslots. Constraints (5) and (6) identify the courses that are assigned to the same day and to consecutive days, respectively. All x_{itd}, y_{ij}, u_{ij}, w_{ij}, and a_{ij} are binary values (7). $a_{ij}, \gamma, \beta,$ and α are positive integers and b_{itd} is a binary parameter.

We implemented this model in IBM CPLEX OPL.

3.2 Heuristic Algorithm

Mathematical modeling requires expert knowledge on modeling and optimization and use of specialized software. Also, the solution times for large sized problems are extensive. Therefore, we developed an efficient heuristic algorithm and implemented it in MS Excel VBA with an easy-to-use interface. Apart from the above-mentioned constraints, the developed tool can handle many practical situations and restrictions. For example, removing a course from the plan if a midterm does need not be scheduled for that course, or fixing the date and time of exams for specific courses, if necessary. After the timetable has been prepared, the tool provides reports on its performance and is ready to announce midterm exam timetable.

The steps of the developed heuristic are as following:

1- Construct the table of common students between all the offered courses.
2- Set the exam date and slots of the courses for which the exam dates are decided to be fixed.
3- Take the two courses which have the highest number of common students.
4- Assign these two courses to their available timeslots which are the class times for the single section courses and the allocated timeslots (stated earlier in midterm timetable mathematical model) for multiple-section courses. Try the first possible timeslots for both courses.
5- Calculate the objective function value (CS, SD, and CD) resulting from this assignment.
6- Repeat steps 4 and 5 for all the possible timeslot assignment alternatives.
7- Choose the best value that will minimize CS, SD, and CD, in respective priority and set the exam date and time for the selected courses.

8- Repeat steps 2 and 7 until all the courses are assigned.
9- Calculate the quality of the heuristic by calculating the objective function value.

This is a construction algorithm which gives priority to courses with a greater number of common students. In Step 3 of the algorithm, such courses are selected and tried for all possible alternative time slots suitable for them in Step 4. For each alternative assignment, the objective function value is calculated (steps 5 6), and the courses are assigned to the best candidate (step 7). This procedure is repeated until all courses are assigned (step 8).

4. Case Study

We performed a computational study and solved the problem for Fall 2023 semester. The exams were to be scheduled during weeks 7-10. Since all midterm grades need to be announced before the end of week 10, only the first three days of week 10 are used for midterms, to allow instructors time for grading. Also, Fridays are holidays for which no exams can be scheduled. As a result, the exams are scheduled over 3 full weeks (excluding Fridays) and the first 3 days of the last week. In total there are 21 days and 6 slots in each day. We used the problem with two data sets. One is a small-sized problem which considered only a single department with 40 courses. The second data set included all the courses of all departments under the College of Engineering (4 departments and 8 undergraduate degree programs) which included 169 courses.

At the beginning of Fall 2023 semester, the department timetabling coordinator prepared the manual exam timetable. In this, 12 students had midterm exams in consecutive slots, and 190 students had exams on consecutive days. However, when the heuristic algorithm was used, no students had exams on consecutive slots or on the same day. Only 3 students had exams on consecutive slots. This is a huge improvement compared to the manual system. Besides the students' and faculty members' satisfaction, the code generates the exam schedule within a few seconds, which would take three days of manual work. The software easily reads the data, processes it, and allows the user to manually modify the data or fix the exam dates.

However, when the departments are considered independently, the solution does not consider the common courses across different departments of the college. Besides common courses, students can also take courses from other departments as program elective courses. As a consequence, although the departmental timetable might seem to be an effective plan, when such courses are considered, there might be a significant number of students with exams on consecutive slots, on the same day or on consecutive days. For this reason, the problem must be solved for the whole college at the same time. In this case, the problem includes a much larger number of courses (169) and students (2076), which increases the complexity. It is not possible to handle this problem manually. When the problem is solved with the heuristic algorithm, again no students have exams on consecutive slots or on the same day and only 22 students have exams in consecutive slots.

During the midterm exam days, classes continue to be taught normally. However, the amount of absenteeism in classes tends to increase because students are stressed about their exams and they do not attend the lectures in order to prepare for their exams on the same or the day. For this reason, it may be beneficial to complete midterm exams in 2 or 3 weeks instead of spreading them over 4 weeks. However, the effect of this may be that more students take the exam in consecutive periods or on consecutive days. It is very easy to test such situations with the developed algorithm. Table 1 shows the results by solving the problem with the exam dates spanning different number of days for both small and large sized problems. It shows the CS, SD, and CD values.

Table 1. Change in the objective function values with respect to the number of exam days

	Single Department			Whole College		
# of Days	CS*	SD*	CD*	CS	SD	CD
12	0	0	175	0	8	771
15	0	0	86	0	0	222
18	0	0	37	0	0	151
22	0	0	3	0	0	44

*CS: Consecutive Slots SD: Same Day CD: Consecutive Days

As it can be seen in this table, even if the exam period is reduced to 2 weeks (12 working days), it is possible to have a timetable with no students having exams on consecutive slots. However, at the college level there will be 8 students who have exams on the same day in non-consecutive slots. The table clearly demonstrates the tradeoff between the number of exam days and the objective functions. Based on this, the college or the university administration can decide on the exam period. This demonstrates that the developed computerized algorithm is an effective tool to prepare timetables as well as perform sensitivity analysis.

5. Conclusions

In this study, we considered the university midterm timetabling problem at the Department and College levels. We developed a mathematical programming formulation together with a heuristic algorithm. The algorithm is implemented in MS Excel VBA which handles the input data easily and provides the solutions in a few seconds. We performed case studies in our institution (at department and college levels) using Fall 2023 data. The results demonstrated that even for the larger problem that includes the whole college, the algorithm generates exam timetables efficiently. It also allows the decision makers to perform sensitivity analysis on problem parameters and obtain insights about the solution characteristics.

As one future study, the algorithm can be modified to schedule final exams. Note that these two problems have several differences. There is no distinction of single and multi-section courses for the final exams. During the final exam periods there are no ongoing courses. The allocated exam slots are longer and the number of spanning exam days is less in the case of final exams. Assigning the exams to classrooms can also be included in the problem. Furthermore, developing a user-friendly graphical user interface and converting the algorithm into a decision support software would be beneficial. As another future study, an exam timetabling program could b e developed for the whole university, with various colleges, and various departments and programs inside each college.

Acknowledgements

Support of Sultan Qaboos University, especially the Department of Mechanical and Industrial Engineering and the College of Engineering, is gratefully acknowledged.

References

Abdul-Rahman S, Sobri NS, Omar MF, Benjamin AM, Ramli R. Graph coloring heuristics for solving examination timetabling problem at Universiti Utara Malaysia. AIP Conf Proc, vol. 1635, 2014. https://doi.org/10.1063/1.4903627.

Abou Kasm O, Mohandes B, Diabat A, El Khatib S. Exam timetabling with allowable conflicts within a time window. Comput Ind Eng 127, 2019. https://doi.org/10.1016/j.cie.2018.11.037.

Al-Yakoob SM, Sherali HD, Al-Jazzaf M. A mixed-integer mathematical modeling approach to exam timetabling. Computational Management Science 7, 2010. https://doi.org/10.1007/s10287-007-0066-8.

Arbaoui T, Boufflet JP, Moukrim A. A matheuristic for exam timetabling. IFAC-PapersOnLine, vol. 49, 2016. https://doi.org/10.1016/j.ifacol.2016.07.701.

Bazari S, Pooya A, Soleimani Fard O, Roozkhosh P. Modeling and solving the problem of scheduling university exams in terms of new constraints on the conflicts of professors' exams and the concurrence of exams with common questions. OPSEARCH 60, 2023. https://doi.org/10.1007/s12597-023-00638-z.

Carlsson M, Ceschia S, Di Gaspero L, Mikkelsen RØ, Schaerf A, Stidsen TJR. Exact and metaheuristic methods for a real-world examination timetabling problem. Journal of Scheduling 26, 2023. https://doi.org/10.1007/s10951-023-00778-6.

Ceschia S, Di Gaspero L, Schaerf A. Educational timetabling: Problems, benchmarks, and state-of-the-art results. Eur J Oper Res 308:1–18, 2023. https://doi.org/10.1016/j.ejor.2022.07.011.

Güler MG, Akyer H, Keskin ME, Döyen A. Examination timetabling problem with scarce resources: a case study. European J of Industrial Engineering 12, 2018. https://doi.org/10.1504/ejie.2018.10016933.

Güler MG, Geçici E. A spreadsheet-based decision support system for examination timetabling. Turkish Journal of Electrical Engineering and Computer Sciences 28:1584–98, 2020. https://doi.org/10.3906/elk-1909-14.

Güler MG, Geçici E, Köroğlu T, Becit E. A web-based decision support system for examination timetabling. Expert Syst Appl 183, 2021. https://doi.org/10.1016/j.eswa.2021.115363.

Kahar MNM, Kendall G. The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. Eur J Oper Res 207, 2010. https://doi.org/10.1016/j.ejor.2010.04.011.

McCollum B, McMullan P, Parkes AJ, Burke EK, Qu R. A new model for automated examination timetabling. Ann Oper Res 194, 2012. https://doi.org/10.1007/s10479-011-0997-x.

MirHassani SA. Improving paper spread in examination timetables using integer programming. Appl Math Comput 179, 2006. https://doi.org/10.1016/j.amc.2005.11.125.

Qu R, Burke EK, McCollum B, Merlot LTG, Lee SY. A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling 12, 2009. https://doi.org/10.1007/s10951-008-0077-5.

Zhu K, Li LD, Li M. Developing an Online Examination Timetabling System Using Artificial Bee Colony Algorithm in Higher Education. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 413 LNICST, 2022. https://doi.org/10.1007/978-3-030-93479-8_7.