Journal of Quality Measurement and Analysis *JQMA* 20(1) 2024, 145-158
e-ISSN: 2600-8602 *http://www.ukm.my/jqma*
https://doi.org/10.17576/jqma.2001.2024.11

FINAL EXAMINATION TIME TABLE SCHEDULING USING INTEGER PROGRAMMING WITH AVERAGE TIME SLOT GAP MINIMIZATION

(Penjadualan Peperiksaan Akhir Menggunakan Pengaturcaraan Integer dengan Purata Pengecilan Jurang Slot Masa)

NUR HIDAYAH ABDUL JAMIL, NOR ALIZA AB RAHMIN* & ATHIRAH NAWAWI

ABSTRACT

The exam of scheduling problem is an academically researched type of scheduling challenge in which exams for a given number of courses are assigned to specific time slots subject to certain constraints. The problem arises when there is a conflict or redundant examination in a time slot of the day. In this research, our aim is to maximize the total amount of students' study time to ensure that all students have sufficient time to study during exam weeks. We used an integer programming approach with an average time slot over the examination for this examination scheduling problem. The algorithm is applied to real data from third-year students of Bachelor of Science in Mathematics with Honours at the Department of Mathematics and Statistics, Faculty Science, UPM during semester 2 session 2021/2022. The result shows that integer programming gives an optimal solution for this complexity. With some improvements to the algorithm, there can be solutions that are better than the manually compiled schedule.

Keywords: scheduling; examination timetabling; integer programming; average time slot over the examination; constraints

ABSTRAK

Isu penjadualan peperiksaan adalah satu jenis cabaran jadualan yang diselidiki secara akademik, di mana peperiksaan untuk beberapa kursus ditetapkan kepada masa - masa tertentu dengan tertakluk kepada beberapa kekangan. Masalah ini timbul apabila terdapat konflik atau pengulangan peperiksaan dalam satu slot masa pada hari tersebut. Dalam penyelidikan ini, matlamat kami adalah untuk memaksimumkan jumlah masa belajar pelajar untuk memastikan semua pelajar memperoleh masa yang mencukupi semasa minggu peperiksaan. Kami menggunakan pendekatan pengaturcaraan integer dengan purata slot masa untuk masalah penjadualan peperiksaan ini. Algoritma ini diaplikasikan ke data sebenar dari pelajar tahun ketiga Ijazah Sarjana Muda Sains Matematik dengan Kepujian pada semester 2 sesi 2021/2022. Hasilnya menunjukkan bahawa pengaturcaraan integer memberikan penyelesaian optimum untuk kompleksiti ini. Dengan beberapa penambahbaikan dalam algoritma, terdapat penyelesaian yang lebih baik daripada jadual yang disusun secara manual.

Kata kunci: penjadualan; ujian penjadualan; pengaturcaraan integer; purata waktu slot bagi ujian penjadualan; kekangan

1. Introduction

Typically, scheduling a timetable is not a new problem for people nowadays. In general, the meaning of timetabling involves deciding when activities or events will take place in time slots but it does not involve assigning to those activities. Real timetabling problems can give examples such as educational timetabling which are courses and examinations, workers timetabling, timetabling of sports events such as football or badminton, timetabling of transportation, and many more (Bhaduri 2009). In this paper, we focus on educational scheduling for examinations. As we know, automated timetabling is very important since it can reduce man costs and deliver practical, ideal solutions for a given set of constraints in a matter of minutes (Bhaduri 2009). However, timetable scheduling is known to be a non-polynomial complete problem (NP-problem) which is very difficult to solve using optimization techniques such as the evolutionary algorithm which is a simplex method (Ghaemi *et al.* 2007)

The challenge in planning the examination timetable scheduling consists of finding the perfect time allocation in which subjects must be arranged around a set of limited time slots to satisfy the hard and soft constraints and also optimize the set of objective functions. The hard constraint is a mandatory constraint that cannot be violated or ignored and overlapped at all while timetabling is being computed (Nanda *et al.* 2012). Then, if we ignored them, the timetable is invalid. For example, the student should not have more than one exam at the same time. The soft constraint is optional if we violate or ignore it while timetabling is being computed, it still can be valid. For example, a gap between one exam to another exam in a week. However, satisfying hard and soft constraints in order to get an effective timetable and feasible solution gives more difficulties.

In this study, the data are taken from real-life examination timetable problems in Universiti Putra Malaysia (UPM) of third-year students in semester two. At the beginning of the semester, they will have technically the same core courses in mathematics subjects. The students are required to register for courses in basic mathematics such as Calculus and Algebra. Furthermore, the students are also required to register for elective courses too. Thus, a challenge to the exam coordinators has been brought out. It is due to the possible conflicting period of the examination time for different courses. There are always some issues with the period gaps between courses, and the number of examinations in a day when the exam schedule is constructed. The solution presented in this paper is an integer programming techniques approach with an average time slot over the examination for this examination scheduling problem to reduce the complications of the problem and eased into the solution. This activity will assist in formulating the optimization model and describing how the algorithm can be adapted to make it applicable to the problem under investigation.

2. Literature Review

It will describe the studies that consider examination schedules as a problem. It comprises two sections which are 2.1 which discussed scheduling in education and in Section 2.2 we will explore various methods used on educational timetabling problems.

2.1. *Scheduling in education*

Scheduling in education is one of the applications for real timetabling problems. There are several categories related to scheduling in education, which are school timetabling, course scheduling, and examination schedules.

Hisham (2022) used Graph Colouring which was aimed to solve timetabling for the final exam for the third year Mathematics students of Universiti Putra Malaysia. Largest Weighted Degree, Recursive Largest First, and DSatur are performed and compared to each other. As we can see in this study, Largest Weighted Degree (LWD) is the best-performing algorithm to obtain the conflict-free exam schedule as a result of the lowest number of steps taken to complete the graph colouring.

2.2. *Educational timetabling problem method*

In this study, we are focusing on using optimization methods in examination schedules which aims to maximize the students' study time between examinations.

Al-Yakoob *et al.* (2010) used a mixed integer programming model (ETM) for exam planning to develop an exam planning problem (ETP) for the Kuwait University (KU) exam planning problem. They discussed two subtopics, namely the ETP, which focuses on exam times and rooms, and the Tutor Assignment Problem (PAP), which regulates the assignment of examiners to exams. The effects achieved far exceed those of the previous manual method in the KU, both in terms of conflict resolution and in terms of completeness, efficiency, and correctness.

Islam *et al.* (2016) proposed the Tabu search method, a meta-heuristic method of creating course and exam schedules for South Eastern University in order to create an acceptable schedule and workable solution. They introduce project notation to solve the schedule generator problem. Therefore, they conclude that the Tabu Search algorithm provides a high-quality solution in a short time and is easy to implement.

In addition, Leite *et al.* (2019) proposed a new variant of the Simulated Annealing (SA) algorithm called FastSA to solve the exam scheduling problem. FastSA competes with the SA algorithm in terms of solution cost, achieving the best average efficiency score with shorter execution times in four out of twelve cases. Therefore, the proposed approach will open new research avenues by developing new algorithms that explore space more efficiently.

3. Methodology

We refer to several past researchers and develop new constraints and objective functions in the model based on previous researchers that can relate to the final examination scheduling problem. Setayesh (2012) stated the constraint on providing enough study time for the students on his project of Examination Scheduling with Days-off Constraints. Next, we also refer to Kalayci and Gungor (2012) that conducted an examination scheduling using Genetic Algorithm (GA) with them one of the hard constraints focused on time which every exam can only be scheduled once in an available time slot. Other than that, we also identify our model on Komijan and Koupaei (2012). They build a mathematical model to set up an examination that according to the difficulty of course scheduling and one of the bounds of the model is each time slot, at most one exam is scheduled and at most three exams can be scheduled on a single day. Moreover, we also refer to Azaiez and Al Sharif (2005) on the weekends' time slot. The model that we construct has been mentioning the next subtopics.

3.1. *Notation*

Table 1: Sets of the model

Set	Description
E	The number of exams to be scheduled;
e	Index of exams ($e \in E$)
T	the number of time slots
t	Index of time slot ($t \in T$)
D	The number of days in scheduled period length
d	Index of days ($d \in D$)
y	Total number of time slots in a day
K	Exam in t time slot in a day
i	1,2,3,4,5

3.2. *Parameters*

Table 2: Parameters of the model

Parameter	Description
T_d	Set of time slots available for day d
s_i	Slack variable

3.3. *Decision variable*

$$x_{edt} = \begin{cases} 1 & \text{if exam } e \text{ is scheduled on day } d \text{ at timeslot } t \\ 0 & \text{otherwise} \end{cases}$$

$$x_{d,e} = \begin{cases} 1 & \text{if day } d \text{ has exam } e \\ 0 & \text{otherwise} \end{cases}$$

$$x_{t,e} = \begin{cases} 1 & \text{if exam } e \text{ is scheduled at timeslot } t \\ 0 & \text{otherwise} \end{cases}$$

$$x_{d} = \begin{cases} 1 & \text{if have exam in day } d \\ 0 & \text{otherwise} \end{cases}$$

3.4. *The model*

The objective function is to maximize the total students' study time between examinations so that the gap between examinations will provide enough study time for the students.

Max

$$\left(\sum_{t+1}^{T} \sum_{e+1}^{E} t.x_{t,e} - \sum_{t=1}^{T} \sum_{e=1}^{E} t.x_{t,e} \right) - 1 + 0s_1 + 0s_2 + 0s_3 + 0s_4 + 0s_5$$

subject to

$$\sum_{d=1}^{D} \sum_{t \in T_d}^{T} x_{edt} = 1 \quad \forall e \in E \tag{1}$$

$$\sum_{e=1}^{E} x_{edt} + s_1 = 1 \quad \forall t \in 56, \forall d \in D \tag{2}$$

$$1 \leqslant t \leqslant T \quad \forall e; e \in 1, 2, 3 \ldots, E \tag{3}$$

$$\sum_{d=1}^{D} x_d + s_2 = 14 \tag{4}$$

$$\sum_{e=1}^{E}\sum_{t\in T_d}^{T} x_{edt} + s_3 = 3 \quad \forall d \tag{5}$$

$$\sum_{t=K}^{K+y-1}\sum_{e=1}^{E} x_{t,e} + s_4 = 2 \tag{6}$$

$$x_{6,e} + x_{7,e} + x_{13,e} + x_{14,e} + s_5 = 0; \quad e = 1, 2, 3 \ldots, E \tag{7}$$

$$x_{edt}, x_d, x_{t,e}, x_{d,e}, s_1, s_2, s_3, s_4, s_5 \in \{0, 1\}$$

Equation (1) restricts that all exams should be scheduled only once in available time slots, T_d. In Eq. (2), each student has to sit at most one examination in each time slot hence, only one exam at each time slot in order to avoid any examination conflicts. Eq. (3) ensures that all examinations must be scheduled among the available time slots. Then, Eq. (4) shows the maximum of examinations that will be held which is 2 weeks (14 days).

The last three constraints are soft constraints in the problems that we have to satisfy in order to get an efficient timetable for final examinations. The first soft constraint shows Eq. (5) which at most three examinations are held in a day. This constraint below ensures that the students will only sit three exams in a day and should not be held at successive time slots and the second constraint, Eq. (6) explains a gap between examinations should be allocated so the student will be able to get enough study and resting time between examinations. The last soft constraint, Eq. (7) shows that no examinations will be conducted during weekends day.

3.5. *Integer programming*

Integer Programming is a method of integer programming in which one of the steps in the algorithm is known as the average of time slot over an exam that we used on this project. This method determined the gap between examinations. It obtains the exact answer of each gap but since it is an average, it might have some additional or subtraction of decimal.

3.6. *Average of time slot over exam*

This average time slot over the exam is used to speed up finding an optimal solution and creates shortcuts to reduce decisional cognitive load. This procedure is solved computationally, where all the algorithm does is move from one solution to another by applying our main action, which is calculating the average of the total time slot over the total number of subjects to know the gap between one subject and another subject, and then following by checking all the constraints that have been listed up until an optimal solution is reached.

Average of time slot over the exam,

$$A_{T,E} = \frac{\text{total number of timeslots available}}{\text{total number of subjects}}$$

new time slot,

$$N_t = t + A_{T,E}$$

subject calculated,

$$E_c = e + 1$$

We round off using two methods: Round off and Round off (Floor). The difference between both methods, Round off (Floor) which gives the output of the integer less than or equal to *x*, and round off, which gives as output of the integer to the nearest x. This is the approach used to compare the objective function.

3.7. *Group*

The courses are classified into groups according to the priority of the subjects.

Table 3: Group of classification

Group	Description
1	• Group of core courses • A mandatory subject to fulfilled the requirement to graduate. • Set as the initial group.
2	• Group of elective that is available and free to register. • The subjects are quite manageable to get an excellent result • Held before other hard subjects in the timetable
3	• All subjects from other departments. • Priority of scheduling will be determined by other faculties themselves.
4	• Elective group from the department of mathematics and statistics. • Register for at least two subjects. • Give some time for the student to get enough quality study to ensure all the topics learned from the beginning managed to catch up.
5	• Elective group on global language. • Subjects that is easy to score and is not very heavy compared to other subjects that required full attention and more work.

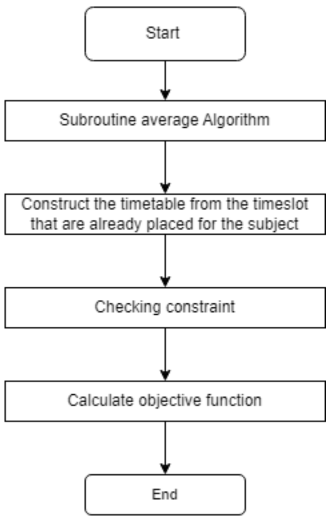

Figure 1: Flow chart of integer programming

3.8. *Algorithm*

Figure 1 illustrates the flow and situation for constructing final examination scheduling beginning with data that is already compiled in one file. The process started with the application of a subroutine average algorithm. This algorithm is used to schedule the subjects into time slots. The process algorithm will be continued until all subjects have been placed. In the last process, The scheduling construct to the timetable and continue with checking all constraints and calculating the objective function on each subject.

This section also introduces an algorithm that is a subroutine average algorithm. The notation used in the algorithm is presented in the following.

Table 4: Notations in the algorithm

g	Group of characteristic ; $g = a, b, c, d, f$
$G[g]$	Array of group
e	Index of exam ; $e \in E$
WY_t	Weekend time slots
WD_1	First time slot on next weekday time slot
t	time slot
N_t	New time slot
$T_{available}$	The total number of time slots available
$A_{T,E}$	Average of time slot per exam
E	The number of exams to be scheduled

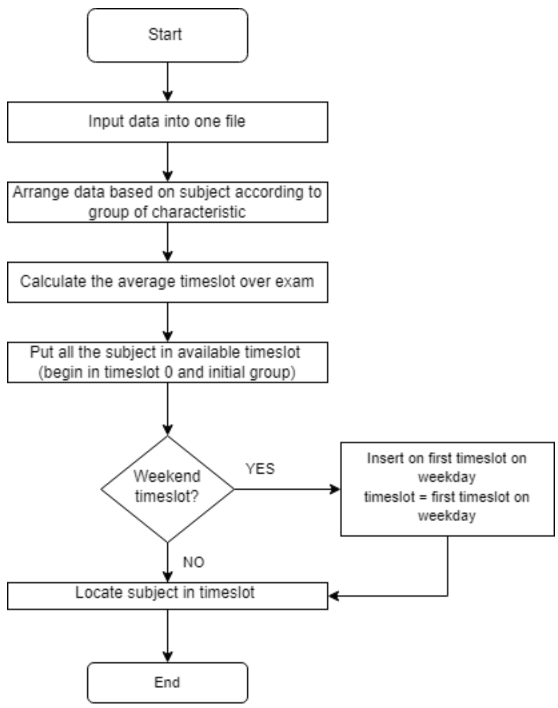

Figure 2: Flow chart of Subroutine average algorithm

The average time slot over the exam algorithm is introduced in Algorithm 1. Step 1, started by inputting data of students such as the time, day, and subjects with their type of courses. Then, the next process which is the data will be assorted according to the group of characteristics of subjects. The step is repeated until the total number of subjects is reached. In step 3, the main activity of the process by calculating the average available time slot over the total subjects to get the fix of gaps maximization. Lastly, the algorithm started to locate each subject starting with the initial group. If the time slot were located in the weekends time slot, then the time slot will be arranged to the first time slot on the next weekday to make sure no examinations were held during weekends. The process repeated until all subjects have been located in the available time slot. Therefore, the timetable of the examination is obtained.

Algorithm 1 Subroutine average

1: $G[g]$ for $g = a, b, c, d, f$ and $n = 1, 2, \ldots, E$ according to t and d
2: Set $g = a$ and $e = 1$
while $G[a] \neq 0$ **do**
    $e = e + 1$
    $G[a] = G[a] + 1$
end while
Repeat process until $g = f$
3: Calculate $A_{T,E}$ for $t_{available}$ and E
4: Set $t = 0$; $a = a + 1$
if $t = WY_t$ **then**
    Move t into WD_1
    $t = WD_1$
else
    $N_t = t + A_{T,E}$
end if
Repeat process until the last subject in a
Proceed next group until last group, f

4. Data

In this study, we used big data which are from Bachelor of Science in Mathematics, Cohort 2018/2022 from Hisham (2022) and data that we collected from the latest student from Bachelor of Science in Mathematics, Cohort 2019/2023.

4.1. *Previous data*

From Hisham (2022) research, Largest Weighted Degree (LWD) is the best-performing algorithm from the three algorithms he used to obtain the conflict-free exam schedule due to the number of steps taken to complete the graph colouring. There are 45 students in total and 27 examinations are to be scheduled.

Table 5: Colour classes using LWD algorithm

Type of courses	**Courses registered**
Red	MTH3406
Blue	MTH3301, MTH4202, MTH4605, PHY3201, LPD2101, LPS2101, LHE3466
Brown	MTH4203, CHM4001, LPC2101, LHE3462, SKP3122
Green	MTH4501, LPJ2101, LPA2102, LHE3410
Pink	FEM2311, LPA2101
Maroon	MTH4604, LHE3408
Yellow	MTH4201
Orange	ECN3161, ACT3211, MGM3211, SSK3207, LHE3404

4.2. *Our data*

The data is collected through a Google Form survey with the participants of 47 students of third-year students from Bachelor of Science in Mathematics with Honours in UPM. The survey is about courses that each student has registered for in Semester 2. The data was collected based on their responses.

Table 6: Courses registered by third-year students of Mathematics cohort 2019/2023 in classified group

Type of courses	Courses registered
1	MTH3406
2	FEM2310, FEM4216, MTH3403, LHE34031
3	ACT3211, ECN3161, ECN3014, MGM3211, PHY3201, SSK3207
4	MTH4201, MTH4202, MTH4203 MTH4501, MTH4604, MTH4605
5	LPD2101, LPG2101, LPM2101 LPS2101, LPT2102

5. Result and Discussion

The algorithm and method are then applied to a final examination problem from Hisham (2022) data which consists of 45 students from the Bachelor of Science in Mathematics with Honours cohort 2018/2022 and compare his result with our approach. We also examined data from the latest third-year student of the Bachelor of Science in Mathematics with Honours cohort 2019/2023, Faculty of Science which consists of 47 students. The result will be getting consists of Experimental result which used data of Hisham (2022) and Computational result which used data from the mathematics student cohort 2019/2023 and compared with the real final examination timetable.

The C programming language was used to obtain the optimal solution and achieve our objective function. The solution provided an optimal timetable that maximized the total gap between exams with the average number of time slots per examination.

5.1. *Experimental Result*

There are four time slots for examinations on Monday until Thursday (0800-1000hours, 1000-1200hours, 1300-1500hours, 1500-1700hours) and three time slots on Friday (0800-1000hours, 1000-1200hours, and 1500-1700hours) since time violated at 1300-1500 hours due to the time that all men Muslims must perform Friday prayer. The examination period will be scheduled for two weeks (14 days) with 38 time slots available and will be held online.

Table 7 shows the optimization result that we obtain using the integer programming method from Hisham (2022) data. The objective function in using this method is 48 hours as the total gap between examination is 19 time slots which indicate 38 hours, and the total break of 11 hours at 1200-1300 hours. Hence, the total gap between examinations is 38 hours + 10 hours gives 48 hours total gap between examinations.

Table 8 shows the result of scheduling the final examination from Hisham (2022) using the LWD algorithm. The objective function of using this method is 56 hours. The total gap between examination is 24 time slots which indicate 48 hours, and the total break of 8 hours at 12000-1300 hours hence, the total gap between examination is 48 hours + 8 hours gives 56 hours total gap between examinations.

Table 7: Final Examination using integer programming

Day	0800-1000	1000-1200	1300-1500	1500-1700
1	MTH3406	MTH3301		LHE3404
2	LHE3466		LHE3408	LHE3410
3	LHE3462		FEM2311	SKP3122
4		ACT3211	ECN3161	CHM4001
5		MGM3211		PHY3201
6				
7				
8	SSK3207	MTH4201		MTH4202
9	MTH4203		MTH4501	MTH4604
10	MTH4605		LPA2101	LPA2102
11		LPC2101	LPD2101	LPJ2101
12		LPS2101		
13				
14				

Table 8: Final Examination using LWD algorithm

Day	0800-1000	1000-1200	1300-1500	1500-1700
1	Green		Pink	
2				
3		Brown		Orange
4				
5		Red	Yellow	
6				
7				
8	Maroon			Blue
9				

The objective function, which is the total gap between examinations obtained from the optimization scheduling result, is 48 hours, and the total gap between examinations obtained from the Hisham (2022) scheduling result is 56 hours. The optimization result gives a slightly lower total gap than the Hisham (2022) timetable. Even though his scheduling result based on the objective function gives a higher amount, it is violated on our constraint 2 which is each student has to sit at most one examination in each time slot hence, only one exam in each time slot in order to avoid any examination conflicts.

5.2. *Computational result*

There are four-time slots reserved for examinations from Monday until Friday. The examination period will be scheduled for two weeks (14 days) with 40 time slots available and will be held online. There are 47 students in total and 22 examinations are to be scheduled.

5.2.1. *Analysis of real data*

Table 9: Final examination scheduling using Round Off(Floor)

Day	0830-1030	1130-1330	1430-1630	2030-2230
1	MTH3406	FEM2310		FEM4136
2		LHE3403		MTH3403
3		ACT3211	ECN3014	
4	ECN3161		MGM3211	
5	PHY3201		SSK3207	MTH4201
6				
7				
8	MTH4202	MTH4203		MTH4501
9		MTH4604		MTH4605
10		LPD2101	LPG2101	
11	LPM2101		LPS2101	
12	LPT2102			
13				
14				

Table 10: Final examination scheduling using Round Off

Day	0830-1030	1130-1330	1430-1630	2030-2230
1	MTH3406		FEM2310	
2	FEM4136	LHE3403		MTH3403
3		ACT3211		ECN3014
4		ECN3161		MGM3211
5	PHY3201		SSK3207	
6				
7				
8	MTH4201		MTH4202	
9	MTH4203	MTH4501		MTH4604
10		MTH4605		LPD2101
11		LPG2101		LPM2101
12	LPS2101		LPST2102	
13				
14				

Using the method of Round Off (Floor), the total gap between examinations is 23 hours while using the method of Round Off, the total gap between examinations is 25 hours.

The result shows that the total gap between the examination and our objective function obtained in method (i) is 112 hours, and the total gap between the examination obtained in method (ii) is 118 hours. By comparing the difference in the total gap between both methods, we can clearly see that method (ii) gives slightly higher total gaps, which more satisfactorily satisfied our goal, which is to maximize the total amount of gap between examinations. As a result, method (ii) is determined to be the best method for obtaining an optimal solution while still maximizing the objective function.

5.3. *Comparison of extracted exam scheduling to real exam scheduling*

Table 11 shows the real exam timetable for 47 students of Bachelor Science of Mathematics with Honour, Faculty Science.

Table 11: Real final examination timetable

Day	0830-1030	1130-1330	1430-1630	2030-2230
1				
2	MTH4604	MTH4605		
3		MTH4203		
4			SSK3207	
5			ACT3211	
6		LPS2101		
7		LPM2101	LPG2101	
8			ECN316 PHY3201	MTH3406
9	MTH4202		LHE3403 FEM4136	
10	MTH4501			MTH4201 MTH3403
11	FEM2310			
12	ECN3014			
13	LPD2101			
14		LPT2102	MGM3211	

Based on the table above, the total gap between examinations obtained from the real exam timetable is 152 hours. The real exam timetable of a third-year student of Bachelor Science in Mathematics with Honour, Faculty Science has generated less appropriately than the methods used to generate final exam timetables, as shown in Table 10, despite the fact that the total gap between the examination or our objective function on real exam timetable was greater than the methods used to generate final exam timetables. It is because examinations will be held on days 6, 7, 13, and 14 (Saturday and Sunday), a weekend that is in contradiction with the soft constraint, S_3 that no examinations will be conducted during weekends. However, the real examination timetable satisfied all the hard constraints that arose from the implemented algorithms.

The real examination timetable is still accepted and feasible, as it only violates the soft constraint. As described in Introduction, the soft constraint is not required to be met because their violation had less of an impact, and it is optional to make a scheduled match what students' ideal final examination schedule is.

6. Conclusion

The aim of the work was to solve the problem of scheduling the final exams. The main goal of scheduling is to maximize the overall gap between exams to give students more time to complete the exam. We used data from 47 third-year mathematics students with honors in the second semester of the 2021/2022 semester. We also used Hisham (2022)'s data to compare with our method to see which method produces the largest overall difference between studies. Consequently, the results of the test are as follows:

- Referring to previous researchers, we developed a model of mathematical formulas and modified it to plan the final exam schedule for third-year students of Bachelor of Science in Mathematics with Honours using the IP.
- We developed a model-based IP algorithm to facilitate the implementation of our solution. Here we used the average of the test time interval as one of the steps to obtain the maximum objective function and solved it using the C programming language.
- We compared the IP method with data from previous studies and the IP result shows a significant difference in relation to the total interval between study weeks.
- The IP method is able to meet the students' wishes with regard to the examination plan.

Acknowledgement

All related information has been provided by the Department of Mathematics and Statistics, UPM. We would like to thank Nor Aliza Ab Rahmin and Athirah Nawawi from the Mathematics and Statistics Department at UPM for the fruitful discussion and contribution to this research.

References

Al-Yakoob S.M., Sherali H.D. & Al-Jazzaf M. 2010. A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science* **7**: 19–46.

Azaiez M.N. & Al Sharif S.S. 2005. A 0-1 goal programming model for nurse scheduling. *Computers & Operations Research* **32**(3): 491–507.

Bhaduri A. 2009. University timetable scheduling using genetic artificial immune network. *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 289–292.

Ghaemi S., Vakili M.T. & Aghagolzadeh A. 2007. Using a genetic algorithm optimizer tool to solve university timetable scheduling problem. *2007 9th International Symposium on Signal Processing and Its Applications*, pp. 1–4.

Hisham F.D.M. 2022. Solving exam scheduling problem using Graph Theory. Unpublished Thesis.

Islam T., Shahriar Z., Perves M.A. & Hasan M. 2016. University timetable generator using tabu search. *Journal of Computer and Communications* **4**(16): 28–37.

Kalayci C. & Gungor A. 2012. A genetic algorithm based examination timetabling model focusing on student success for the case of the college of engineering at Pamukkale University, Turkey. *Gazi University Journal of Science* **25**(1): 137–153.

Komijan A.R. & Koupaei M.N. 2012. A multi-attribute decision-making and mathematical model for university examination timetabling. *Journal of Basic and Applied Scientific Research* **2**(10): 10258–10262.

Leite N., Melício F. & Rosa A.C. 2019. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications* **122**: 137–151.

Nanda A., Pai M.P. & Gole A. 2012. An algorithm to automatically generate schedule for school lectures using a heuristic approach. *International Journal of Machine Learning and Computing* **2**(4): 492–495.

Setayesh A. 2012. Examination scheduling with days-off constraints. Master's thesis, Dalhousie University.

Department of Mathematics and Statistics
Faculty of Science
Universiti Putra Malaysia
43400 UPM Serdang
Selangor, MALAYSIA
*E-mail: 203560@student.upm.edu.my, aliza@upm.edu.my**, *athirah@upm.edu.my*

Received: 5 May 2023
Accepted: 18 August 2023

*Corresponding author