Menemui Matematik (Discovering Mathematics) 47(2) (2025) 28-42

Menemui Matematik
(Discovering Mathematics)

journal homepage: https://persama.org.my/dismath/home

Minimizing Conflicts in Final Examination Problem Using Heuristic Procedure

Arina Yasmin Fauzi[1], Nor Aliza Ab Rahmin[2*] and Athirah Nawawi[3]
[1,2,3]*Department of Mathematics and Statistics, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor*

aliza@upm.edu.my
*Corresponding author

Received:25 May 2025
Accepted:13 August 2025

ABSTRACT

Examination timetabling is a complex problem involving various constraints, where exams must be scheduled in specific time slots while minimizing conflicts. The challenge arises when students take several exams within a short period, which would truncate their preparation time and potentially affect performance. This study introduces a heuristic-based approach to optimize final examination scheduling by prioritizing courses with higher student enrolment. The heuristic algorithm operates by first sorting courses based on enrolment size and then assigning them to available time slots in a manner that avoids overlap for students enrolled in multiple courses. This method ensures that high-impact exams are placed earlier in the scheduling process, reducing the likelihood of critical conflicts. The heuristic method is evaluated by comparing its performance with the Graph Colouring and Integer Linear Programming methods. In addition, modifications to the Integer Linear Programming model are proposed to maximize scheduling flexibility and reduce exam conflicts. The model is applied to real third-year student data of the Bachelor of Science in Mathematics and Mathematics with Education at Universiti Putra Malaysia. The findings indicate that the heuristic approach effectively minimizes scheduling conflicts while having a balanced examination distribution, providing a fairer and more efficient final examination schedule.

Keywords: Examination timetabling, Heuristic procedure, Scheduling optimization, Conflict minimization, Constraints.

INTRODUCTION

Optimization is crucial in various real-life applications, including final examination timetabling. Scheduling examinations in universities is challenging due to the multiple constraints that must be considered. The examination timetable must be constructed to minimize conflicts, prevent students from sitting multiple exams within short timeframes and ensure an equitable distribution of exams across available slots (Abou Kasm et al., 2019). Poorly scheduled examination timetables can lead to high cognitive load and stress among students, impacting their academic performance. Therefore, developing effective and practical optimization strategies is essential to produce examination timetables that are both efficient and student friendly.

Final examination scheduling is a nondeterministic polynomial time (NP) hard problem that involves assigning a set of exams to a fixed number of time slots while satisfying constraints (Burke et al., 1995). The complexity arises from the need to adhere to hard constraints, such as restricting all exams must be scheduled once in available time slots. Hard constraint is mandatory and must be strictly adhered to and violating this constraint results in infeasible timetables that cannot be implemented. Meanwhile, soft constraint, for example, students should sit at most three exams in a day. Traditional scheduling methods may face challenges in efficiency reducing conflicts due to the number of students, courses and constraints, leading to unfair scheduling.

This study aims to maximize the total gap between examinations by developing a heuristic procedure to ensure fair examinations distribution. The data are taken from real-life examination timetable problems from third-year students in semester two of Universiti Putra Malaysia (UPM). This paper presented a heuristic procedure to final examination timetable by prioritizing courses with higher student enrolments to reduce the conflicts. The heuristic procedure is tested against Graph Colouring and Integer Linear Programming methods from previous research, (Hisham 2022) and (Jamil 2023). Additionally, modifications to the ILP model are introduced to enhance scheduling flexibility and reduce exam conflicts. This process will aid in developing an optimized scheduling model while adapting the algorithm to effectively address the challenges of the examination timetabling problem.

OBJECTIVES

- To test the algorithms of the Heuristics approach for solving examination timetables using real data.
- To modify the Integer Linear Programming (ILP) model of the final examination timetable schedules using real data.

UNIVERSITY COURSE TIMETABLE PROBLEMS

University course timetabling is a complex challenge that involves scheduling courses at specific times and locations while addressing the varied needs of students, instructors and available instructional resources. Colajanni (2020) applied Integer Linear Programming (ILP) to create a timetable that ensure the courses, teachers and classrooms are scheduled without conflicts or overlaps. By employing an Integer Linear Programming (ILP) approach to a real-world scenario and comparing it with literature cases, the study demonstrated the effectiveness of ILP in addressing scheduling conflicts and efficiently managing resources. In the same year, Gozali and Fujimura (2020) faces challenges in students sectioning, where individual course preferences exponentially increase scheduling complexity. To address this, the Multi-Depth Genetic Algorithm (MDGA) was introduced to utilize advanced mutation techniques and a multi-layered approach to enhance scheduling efficiency. This methodology represents the potential to address the complexities of student sectioning and enhance the effectiveness of university course timetabling solutions.

Chen et al. (2022) employ heuristic ordering at Universiti Malaysia Sarawak (UNIMAS) to resolve extended class hours until 9 p.m. and unallocated courses. To improve feasibility, the researchers employ heuristic orderings to generate a feasible initial solution, prioritizing courses based on criteria such as the largest degree and enrolment. By using perturbation approaches, which modify the initial solution to minimize unallocated courses, they improve the scheduling process even further. Recent studies, Gu et al. (2025) faces challenges in effectively allocating courses while minimizing conflicts and optimizing resource use. To overcome these issues, integer programming particularly with CPLEX proving effective and recent advancements focus on hybrid models integrating machine learning for adaptability.

EXAMINATION TIMETABLE PROBLEMS

Examination timetabling is challenging due to the complexity of constraints and requirements. Unlike course scheduling, it must avoid overlapping exams and minimize back-to-back exams to allow sufficient study time. Hisham (2022) examined the timetabling challenges specific to scheduling the final examination for third-year Mathematics students of Universiti Putra Malaysia. The study performed Graph Colouring techniques, testing and comparing three algorithms including Largest Weighted Degree (LWD), Recursive Largest First (RLF), and DSatur to construct conflict-free examination timetables. Among these methods, LWD produced the best-performing algorithm to obtain a conflict-free examination schedule as the evaluation metrics relied on the number of steps required to colour the graph, with fewer steps directly corresponding to higher efficiency. Jamil (2023) introduced an Integer Linear Programming (ILP) model to optimize exam timetabling by maximizing study gaps between exams. The study emphasized that fair exam scheduling improves student's academic performance while addressing logistical constraints.

METHODOLOGY

We use the ILP model from previous studies as a base and implement it within our method to achieve the intended objectives and constraints. ILP is known to be an effective timetabling approach due to its structured formulation of variables, constraints, and objective functions. By minimizing or maximizing a specified objective while ensuring all constraints are satisfied, it makes it possible to generate optimal solutions. Thus, this research utilizes the model proposed by Jamil (2023) as the foundation for our heuristic procedure to enhance the fairness of examination timetabling. This integration ensures theoretical consistency while enhancing the fairness and practicality of examination timetabling in real-world settings.

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	Total number of timeslots
t	Index of timeslot $(t \in T)$
D	The number of days in the scheduled period length
d	Index of days $(d \in D)$
y	Total number of timeslots in a day
K	Exam in t timeslot in a day
i	Number of slack variables

PARAMETERS

Table 2: Parameters of the model

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

DECISION VARIABLE

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

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

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

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

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

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

Subject to

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

$$\sum_{e=1}^{E} x_{edt} + s_1 = 1, \qquad \forall t \in \{0,1,2,\dots,56\}, \qquad \forall d \in D \tag{2}$$

$$1 \leq t \leq T \qquad \forall e; e \in 1,2,3,\dots,E \tag{3}$$

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

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

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

$$x_{6,e} + x_{7,e} + x_{13,e} + x_{14,e} + s_5 = 0; \qquad e = 1,2,3, \dots, E \qquad (7)$$

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

Equations (1) to (4) represent the hard constraints of this model. Equation (1) restricts all exams should be scheduled once in available time slots. For Eq. (2), each student must sit at most one examination in each timeslot hence, one exam in each timeslot 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 other three constraints in the model are soft constraints. The first soft constraint is Eq. (5) states that each student should sit at most three exams in a day. It means that there's at most three examinations are held in a day and should be held at successive time slots. Next, Eq. (6) explains a gap between examinations should be allocated so the students will be able to have rest time between examinations and get enough study. The last constraint which is Eq. (7) in this model shows that all exams cannot be held on weekends. This model structure is adapted from Jamil (2023), with modifications to enhance fairness and reduce student stress during the examination period.

HEURISTIC PROCEDURE

Heuristics can be defined as methods that aid in problem-solving that rely on rules and strategies to facilitate decision-making. In the context of timetabling, heuristic approaches simplify decision-making by generating near-optimal solutions within a reasonable time frame, making them particularly useful for complex scheduling problems. Loo at al. (1986) demonstrated the use of heuristics by developing a computerized system known as the Timetable Scheduler (TTS). Gunawan (2017) further identified several widely used heuristic techniques in timetabling, including greedy algorithms, genetic algorithms, simulated annealing, and tabu search. These methods enable schedulers to create workable schedules by exploring various combinations of resources and constraints iteratively.

LARGEST ENROLMENT HEURISTIC

The number of students enrolled in an event is the major measure of scheduling complexity. The larger the number of students enrolled, the harder it will be to organize exams due to greater chances of conflicts and lower availability of suitable timeslots and venues. This complexity arises from the need to accommodate more students while satisfying various academic and logistical constraints. Pillay (2014) highlighted that as student enrolment increases, the difficulty of generating feasible timetables also rises significantly. Therefore, more advanced heuristic or hyper-heuristic methods are required to effectively manage these timetable conflicts.

ALGORITHM

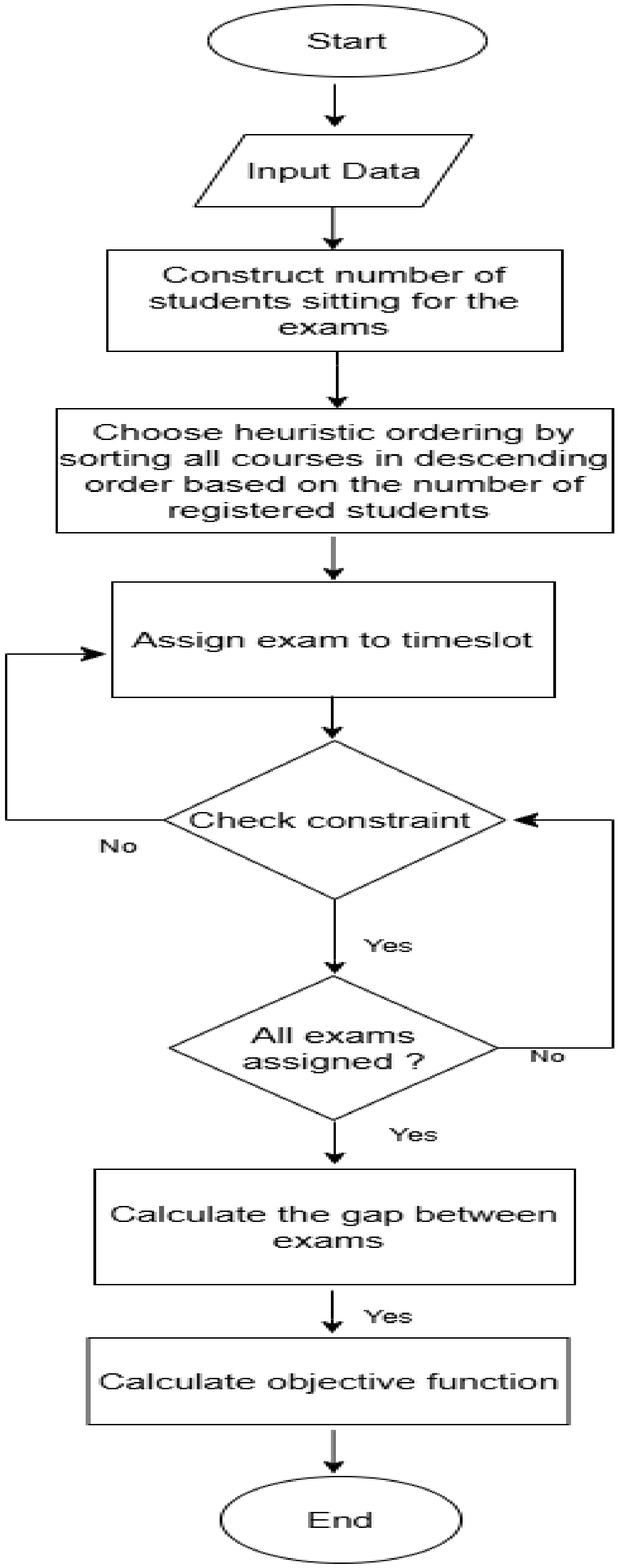

Figure 1: Flowchart of Heuristic Procedure

Figure 1 illustrates the flow and process for final examination scheduling. The process starts with inputting data in one file. The next step is constructing the number of students sitting for each exam. the algorithm begins by sorting all courses in descending order based on the number of registered students. This sorted list represents the heuristic ordering that determines the priority of exams during the scheduling process. Following heuristic ordering, the exams will be assigned to the available timeslots. Each exam is tentatively placed into an available timeslot while considering various constraints. By assigning exams with the highest enrolment first and resolving conflicts dynamically during assignment, the algorithm aims to reduce peak conflict loads early in the scheduling process.

Once an exam is assigned, the constraints are checked to verify whether the placement is feasible. If the constraints are violated, the process returns to the assigned exam timeslot, and adjustments are made accordingly. If the constraints are satisfied, the next step determines whether all exams have been assigned. If not, the process continues assigning and checking constraints until all exams are successfully scheduled.

The steps continued by calculating the gap between exams, ensuring students have adequate breaks between exams. Finally, the objective function is calculated by calculating the total gap between examinations. Therefore, the timetable of the examination is obtained.

DATA

In this study, we used small data which are from Bachelor of Science in Mathematics, Cohort 2018/2022 from Hisham (2022), big data from Bachelor of Science in Mathematics, Cohort 2019/2023 and the data that we collected from Bachelor of Science in Mathematics with Honours and Bachelor of Science in Mathematic with Education (Honours), cohort 2021/2025. A small dataset was used in the initial phase to make it easy to validate and debug the proposed model. With fewer entries, it is easier to manually validate the accuracy of the schedule and ensure all constraints are enacted precisely. This approach allows for efficient identification and correction of errors before scaling the model to a larger dataset.

PREVIOUS DATA

The previous data used in this research were obtained from two different studies, which are from Hisham (2022) and Jamil (2023). Each study focused on different methods to solve the final examination problem. Hisham (2022) used the graph colouring method while Jamil (2023) applied Integer Programming.

FIRST DATA

From Hisham (2022) research, the Largest Weighted Degree (LWD) is the best-performing algorithm of the three algorithms he used to obtain the conflict-free exam schedule due to the number of steps taken to complete the graph colouring which his goal to build a conflict-free timetable. They are 5 students for small data in total and 11 examinations are to be scheduled.

Table 3: Courses registered by five respondents randomly

Students	Courses registered
1	MTH3406, MTH4203, MTH4501, PHY3201
2	MTH3406, MTH4202, MTH4203, ECN3161, FEM2311
3	MTH3406, MTH4202, MTH4501
4	MTH3406, MTH4501, ACT3211
5	MTH3406, MTH4201, MTH4604, CHM4001

Then, the registered courses of all five respondents are listed according to their classification in the LWD algorithm's colour classes.

Table 4: Colour classes using LWD algorithms

Colour Classes	Courses
Red	MTH3406
Blue	MTH4202, PHY3201
Brown	MTH4203, CHM4001
Green	MTH4501
Pink	FEM2311
Maroon	MTH4604
Yellow	MTH4201
Orange	ECN3161, ACT3211

SECOND DATA

Jamil (2023) employed Integer Programming to address the final ex amination scheduling problem. The researcher calculated the average number of timeslots per examination using two different rounding methods and compared the results. The first method is round off (floor), (method i), which gives as output the greatest integer less than or equal to x. In contrast, the second method is the round off (method ii), which rounds off the number to the nearest integer. The findings indicated that the nearest integer rounding method resulted in a more balanced distribution of the time frame between examinations. From Jamil (2023) research, the courses are divided into two categories, which are core courses(mandatory) and elective courses, which are divided into four groups (G1, G2, G3, G4). There are 47 students in total, and 22 exams are to be scheduled.

Table 5: Courses classes using group

Group	Courses registered
1	MTH3406
2	FEM2310, FEM4136, LHE3403, MTH3403
3	ACT3211, ECN3014, ECN3161, MGM3211, PHY3201, SSK3207
4	MTH4201, MTH4202, MTH4203, MTH4501, MTH4604, MTH4605
5	LPD2101, LPG2101, LPM2101, LPS2101, LPT2102

SORT BY THE EXAM ENROLMENT

The measure of the exams with the number of students sitting for the exams from small data from Hisham (2022) is scheduling approach that prioritizes those with the highest enrolment. To archive this, the exam enrolment data is sorted in descending order based on the number of students registered for each examination. The exams are arranged based on enrolment size by focusing on the number of students enrolled in each examination. Then, the exam sorted in descending order based on the number of students registered for each examination.

Table 6: Sorted list of courses based on the exam enrolment

Courses Code	Exam Enrolment
MTH3406	5
MTH4501	3
MTH4203	2
MTH4202	2
ECN3161	1
FEM2311	1
PHY3201	1
MTH4201	1
MTH4604	1
CHM4001	1
ACT3211	1

WEIGHTED TIMESLOT ALLOCATION

To determine student preferences, students were directly asked to identify their most and least preferred examination times. A weighting system was developed based on the collected responses, assigning higher weights to the most preferred timeslots. The examinations were scheduled in descending order of preference, from the most preferred to the least preferred timeslots.

Table 7: Weighted Values for Timeslots Based on Student Preferences

Day	8a.m - 10a.m	10a.m - 12p.m	1p.m - 3p.m	3p.m - 5p.m
1	39	50	29	10
2	38	49	28	7
3	48	37	27	6
4	47	36	26	46

5	35	29	0	4
6	0	0	0	0
7	0	0	0	0
8	45	34	24	44
9	33	23	43	4
10	32	42	22	3
11	41	25	21	2
12	30	40	0	1
13	0	0	0	0
14	0	0	0	0

COMPARISON OF SMALL DATA

We use small subset of the original dataset reported in Hisham (2022), first data, which is randomly selected 5 out of 45 students. Like Jamil (2023)'s approach, which used a small subset of data derived from Hisham (2022)'s research. This allows us to compare Hisham (2022)'s and Jamil (2023)'s method and our proposed heuristic procedure, which are exam enrolment and weighted timeslot allocation. Thus, we aim to demonstrate and illustrate a comparison using a small sample of data from Hisham (2022) and Jamil (2023) to access the gap in the examination schedule results through their procedure with our results obtained using heuristic approach. According to the result of using the heuristic procedure on five students in Hisham (2022) small data, we have seen that heuristic procedure which is exam enrolment (Without Weighted) maximizes the total gap between examinations hence it achieves our objective function.

Table 8: Comparison of Methods using Small Data

Method	Total gap of exam in examination weeks (hours)
Graph Colouring	56
Integer Programming	75
Heuristic Procedure	86
Heuristic Procedure with Weighted Allocation	79

COMPARISON OF REAL DATA

We use the second data from previous research which is Jamil (2023) to compare the Integer Programming Method with our heuristic procedure. Each day from Monday to Friday consists of four examination timeslots: 8:30–10:30 AM, 11:30 AM–1:30 PM, 2:30–4:30 PM, and 8:30–10:30 PM.

Table 9: Final Examination using Integer Programming

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
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				

Table 10: Final Examination using Heuristic Procedure

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
1	MTH3406	MTH4604		MTH4203
2				
3	MTH4501		FEM4136	ECN3161
4	MTH4201	ACT3211		MTH4202
5			MGM3211	MGM3211
6				
7				
8	LPD2101		LHE3403	ECN3014
9		FEM2310	MTH3403	
10			PHY3201	LPG2101
11	LPM2101			
12	LPT2102	SSK3207		LPS2101
13				
14				

The objective function, which is the total gap between examinations obtained from our method is 124 hours, and the total gap between examinations obtained from the Jamil (2023) scheduling result is 118 hours. Our result provides a longer total gap (124 hours) than the Jamil (2023) (118 hours) timetable.

EXTRACTION OF GENERATED EXAM TIMETABLE

We selected one student randomly from 47 students in Jamil (2023) data and extracted the registered courses to analyse the impact of examination scheduling on individual students.

Table 11: Final Examination Schedule of 200734 students on Heuristic Procedure

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
1	MTH3406			
2				

3				
4	MTH4201	ACT3211		

We obtained the total gap between examinations for this student is 41 hours. Despite this gap, a significant scheduling issue was identified: the student has back-to-back examinations for MTH4201 and ACT3211, meaning these exams are scheduled consecutively without a sufficient break in between as it does not provide the student to rest and prepare adequately for the next exam. This limitation suggests the need for further refinement of the scheduling model to ensure that students are not required to sit for consecutive exams without sufficient time for rest and preparation.

MODEL MODIFICATION

While the heuristic procedure aimed to maximize the overall gap between examinations, it did not guarantee a fair distribution of breaks for students, particularly those facing consecutive exams. Consequently, some students experienced back-to-back examinations without sufficient time for rest and preparation. To address the limitation identified in the model, a modification has been introduced to improve the fairness and efficiency of the examination scheduling process. This adjustment involves implementing a new constraint that prevents back-to-back exams by ensuring a minimum gap between exams with similar or higher enrolment.

NEW CONSTRAINT

The newly implemented constraint ensures at least one timeslot gap between exams with similar or higher student enrolment. The constraint is formulated as follows:

$$x_{t_1,e_1} + x_{t_2,e_2} \leq 1 + \left(t_2 - t_1 \geq g_{min}\right) \tag{8}$$

Equation (8) introduces a soft constraint that aims to create a reasonable gap between two exams with a high number of student enrolments. In simple terms, if two exams e_1 and e_2 are scheduled in timeslots t_1 and t_2, then t_2 should be at least one slot after t_1. This is because exams with many students are more likely to have overlapping candidates. To address this, the algorithm checks all exam pairs and compares how many students are enrolled in each. If the enrolments are similar or high, it tries to spread the exams apart, so students don't have to sit for back-to-back papers. Although the model cannot always guarantee this in every situation, it tries to meet this condition as much as possible to reduce student stress and improve preparation time.

RESULT FOR MODIFIED MODEL

After incorporating the new constraint into the model, the final examination timetable was re-evaluated to assess its impact on scheduling efficiency and student fairness.

Table 12: Final Examination Schedule of 200734 students on Heuristic Procedure after model modification

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
1	MTH3406			
2				

3		MTH4201		
4	ACT3211			

The results indicate an improvement in examination scheduling after implementing the model modification. Before the modification, the student had back to-back examinations for MTH4201 and ACT3211 as observed in Table 10. After implementing the new constraint to enforce a minimum gap between exams with overlapping students, the updated schedule, shown in Table 11 after modification, demonstrates a more balanced distribution of exams. The total gap timeslot between examinations is 20 hours. The total break time over four days amounts to 18 hours. Thus, the total gap between examinations is calculated as: 20 hours (timeslot gap) + 18 hours (break time) = 38 hours. Although the total gap between examinations was slightly reduced to 38 hours after the modification, compared to 40 hours previously, this modification effectively prevents back-to-back examinations, allowing students adequate time for rest and preparation between examinations.

RESULT AND DISCUSSION

We apply the modified model to our collected dataset by implementing it in Python. We collected the data through a Google Form survey from third-year students in the Bachelor of Science in Mathematics with Honours and the Bachelor of Science in Mathematics with Education (Honours) programs at the Faculty of Science, Universiti Putra Malaysia (UPM). There are 21 participants, and 6 examinations are to be scheduled from a Bachelor of Science in Mathematics with Education (Honours) while there are 46 participants in total and 27 examinations are to be scheduled from Bachelor of Science in Mathematics with Honours.

Table 13: Final Examination Scheduling for Bachelor of Science in Mathematics with Education (Honours)

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
1				
2	STE4480		MTH3602	
3		FCE3101		
4				
5				FCE3001
6				
7				
8		MTH4501		
9				
10				
11	MTH4203			

The total gap between examinations is 31 timeslots which indicates 62 hours. The final exam spans 10 days, with a total breaktime of 54 hours. Consequently, the total gap between examinations is calculated as 62 hours + 54 hours, resulting in 116 hours of total gap examinations.

Table 14: Final Examination Scheduling for Bachelor of Science in Mathematics with Honours

Day	8:30–10:30 AM	11:30 AM–1:30 PM	2:30-4:30 PM	8:30-10:30 PM
1	MTH3301		MTH3602	MTH4501
2	MTS3701	MTS4503	MTS4211	

3	MTS3602	MTS4102		CSC4500
4			STS3404	STS4407
5	STS4408	MTS4106	ACN3102	
6				
7				
8			LHE3408	MTS4203
9	LPK2104	BBM3402	LPJ2104	
10	LPA2104	LPS2104		MTS4107
11	MTS3101	MGM3211	ECN3010	
12			STS3402	MTS3203

The total gap between examinations is 21 timeslots, equal to 42 hours. The final exam spans 12days, with a total break time of 72 hours. Consequently, the total gap between examinations is calculated as 42 hours + 72 hours, resulting in 114 hours of total gap examinations.

CONCLUSION

This research successfully applied and tested the heuristic approach using real student data from Bachelor of Science in Mathematics with Honours and the Bachelor of Science in Mathematics with Education (Honours). We also used Hisham (2022)'s small data and Jamil (2023)'s real data to compare their method with our method to see which method produces maximum total gap between examinations. By comparing our method with previous methods, the results demonstrated that our method achieved better performance in terms of maximizing the gap between examinations. Additionally, modifications were introduced to the Integer Linear Programming (ILP) model as we noticed conflicts between examinations in which students had to do back-to-back examinations. Equation (8) was introduced as a soft constraint to help reduce the chances of students having back-to-back exams. When applied to real data, the modified ILP model effectively addressed scheduling constraints, contributing to a more structured and efficient timetable. Overall, the findings confirm that both the heuristic approach and the enhanced ILP model offer viable solutions for optimizing final examination scheduling, aligning with the study's objectives of modifying and testing these methods using real student data.

REFERENCES

Chen, M. C., Sze, S. N., Goh, S. L., and Lau, S. P. (2022), Investigation of heuristic orderings with a perturbation for finding feasibility in solving real- world university course timetabling problem. In *2022 International Conference on Digital Transformation and Intelligence (ICDI)*: 168–173.

Colajanni, G. (2020), An integer programming formulation for university course timetabling. AIRO Springer Series/AIRO Springer Series: 219–231

Gozali, A. A. and Fujimura, S. (2020), Solving university course timetabling problem using multi-depth genetic algorithm: Solving uctp using mdga. SHS Web of Conferences, 77:01001.

Gu, X., Krish, M., Sohail, S., Thakur, S., Sabrina, F., and Fan, Z. (2025), From integer programming to machine learning: A technical review on solving university timetabling problems. A Technical Review on Solving University Timetabling Problems. Computation, **13(1)**:10.

Hisham, F. (2022). Solving exam scheduling problem using graph theory. B.s. thesis, Universiti Putra Malaysia.

Jamil, N. H. B. A. (2023), Final examination timetable scheduling using integer programming with average time slot gap minimization. *Journal of Quality Measurement and Analysis*, **20(1)**:145–158.

Pillay, N. (2014), A review of hyper-heuristics for educational timetabling. Annals of Operations Research, **239(1)**:3–38.

Burke, E. K., Elliman, D., & Weare, R. F. (1995), The Automation of the Timetabling Process in Higher Education. **23(4)**, 353–362.

Abou Kasm, O., Mohandes, B., Diabat, A., & El Khatib, S. (2019). Exam timetabling with allowable conflicts within a time window. Computers & Industrial Engineering, **127**: 263–273.

Loo, E., Goh, T., and Ong, H. (1986). A heuristic approach to scheduling university timetables. Computers Education, **10(3)**.