ENHANCED KNOWLEDGE IN SCIENCES AND TECHNOLOGY
e-ISSN: 2773-6385

EKST

Vol. 6 No. 1 (2026) 491-500
https://publisher.uthm.edu.my/periodicals/index.php/ekst

Comparative Study of Graph Colouring and Simulated Annealing for Examination Timetabling at UTHM Pagoh

Chang Chu Yun[1], Wan Nor Munirah Ariffin[1]*

[1] *Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, UTHM Kampus Cawangan Pagoh, Hab Pendidikan Tinggi Pagoh, KM1, Jalan Panchor, 86400 Pagoh, Muar, Johor, MALAYSIA.*

*Corresponding Author: normunirah@uthm.edu.my
DOI: https://doi.org/10.30880/ekst.2026.06.01.055

Article Info

Received: 4 January 2026
Accepted: 6 June 2026
Available online: 9 July 2026

Keywords

Examination Timetabling, Graph Colouring, Simulated Annealing, Heuristic Comparison, Scheduling Optimization

Abstract

The Examination Timetabling Problem (ETP) represents a significant challenge in academic administration. It requires the efficient allocation of exams to limited number of time slots while avoiding conflicts where a student is scheduled for two exams at once. Although various automated heuristics exist to solve this problem, but few studies directly compare their performance under identical conditions. This gap in the literature makes it difficult for institutions to select the most appropriate method. This study aims to fill that gap by fairly comparing two popular methods, namely as Graph Colouring (GC) and Simulated Annealing (SA). This study uses a real dataset from Universiti Tun Hussein Onn Malaysia. The dataset consist of 34 examinations with 941 students in five academic programmes. Both algorithms successfully generated conflict-free timetables. The findings show that the two methods have very different strengths. The results show that Graph Colouring performs well in terms of speed and compactness, and can schedule all exams in 0.0056 seconds with 6 time slots. However, this efficiency generates high load imbalance, with the highest number of 387 students in one session. In contrast, SA is more focus on balance, decreasing the peak occupancy by 75.7% to 94 students per slot, but needs 16 slots and take more computation time. According to this analysis, Graph Colouring is recommended as the more practical choice in this study, providing a fast and compact solution that aligns with the goal of minimizing the examination period. This study offers evidence-based guidelines to help administrators in selecting the most suitable timetabling approach that fits their specific context.

1. Introduction

Timetabling is a crucial and complex administrative activity that is fundamental in the operation of educational institutions. The Examination Timetabling Problem (ETP) is one of the hardest scheduling problems. The ETP involves assigning a set of examinations to a limited number of time slots and venues while satisfying multiple hard and soft constraints [1]. The procedure should be strictly guided by a set of rules, which are called hard constraints. The key hard constraints are avoiding conflicts in which a student may have several exams at the same time and ensuring the number of students in any room does not exceed its capacity [2][3]. Other rules that are preferable but not mandatory, referred to as soft constraints, such as spacing out exams among the same student cohort to reduce fatigue. The complexity of ETP increases with institutional size, growing student numbers, and modular course structures, making fully optimal timetables challenging to achieve in practice [4][5].

Examination timetables have become difficult to build manually. Manual scheduling is time-consuming due to the sheer scale of the problem, the large number of conflicting constraints, and is also subject to variability across departments, particularly when dealing with large cohorts of students [6][7]. In contrast, the adoption of automated solutions is critical. These systems can produce practical, high-quality timetables effectively, and thus minimize administrative burden and enhancing accuracy [8].

The ETP is recognized as an NP-hard combinatorial optimization problem, where the objective is to select the most suitable solution from a huge number of options [9]. The complexity of the problem also rises exponentially with the computational demands, limiting the practical application of exact optimization methods to smaller or fewer issues constrained [1]. As a result, heuristic and other automated scheduling approaches are widely adopted in practice to generate feasible, high-quality timetables within a reasonable amount of computational time [10].

Graph Colouring (GC) and Simulated Annealing (SA) are two popular methods that have been used as heuristic-based approaches for examination timetabling. Graph Colouring, a fundamental concept in graph theory, has been extensively applied as a heuristic where conflict avoidance is the primary concern [11]. Greedy GC algorithms are especially valued for their simplicity and high-performance in terms of time to produce conflict-free schedules [12]. On the other hand, Simulated Annealing is a single-solution metaheuristic derived from statistical mechanics and inspired by physical annealing processes in metallurgy [13]. It is widely applied to combinatorial optimization problems including examination timetabling due to its ability to deal with complex constraint structures and avoid local optima [14].

Recent literature still shows the applicability of the two methods. of Graph Colouring has demonstrated its practical efficiency in applications such as the scheduling of exams in various university departments [12]. The effectiveness of greedy graph colouring algorithms has also been further highlighted, which have shown significant reductions in time and computational resources and an ability to satisfy hard and soft constraints [15]. For Simulated Annealing, recent studies have developed techniques of standardized timetabling benchmarks [16]. Other implementations have been characterized by their simplicity and adaptability to domain-specific constraints [14]. The ITC 2007 benchmark has also been introduced with a single-stage SA method with feature-based parameter tuning, which proves that SA is applicable to standardized timetable challenges [17].

As highlighted in recent literature, there is a knowledge gap in how different heuristic techniques perform when evaluated on the same problem with the same constraints [8]. This gap is essential for NP-hard timetabling problems where the main challenge is to select an effective solution among numerous alternatives [1][18]. Without such comparisons, it is difficult to have a clear understanding of the relative strengths and weaknesses of each approach and their applicability to the fulfilment of specific operational priorities in real-life examination timetabling situations.

To fill this gap, this study conducts a controlled empirical comparison of the Greedy Graph Colouring and Simulated Annealing algorithms. The study uses the two methods on the same dataset of the Faculty of Applied Sciences and Technology at Universiti Tun Hussein Onn Malaysia (UTHM), Pagoh Campus, in the first semester of the 2024/2025 academic year. The main objectives are: (1) to develop GC and SA models to solve the ETP, (2) to produce exam timetables using both models under the same constraints and dataset, and (3) to compare the performance of both models in terms of such measures as conflict avoidance, feasibility compliance, schedule compactness, student load distribution, and computational efficiency.

The findings will be utilized to offer practical recommendations to academic administrators to determine the trade-offs between algorithmic speed and schedule quality to choose an appropriate timetabling strategy based on the requirements of an institution.

2. Materials and Methods

This section describes the experimental setup that includes the dataset, problem formulation, implementation of the two heuristic models, and the evaluation metrics. The methodology ensures a fair and controlled comparison between Graph Colouring and Simulated Annealing.

2.1 Data Preparation and Dataset Description

The study is based on a real administrative data of Universiti Tun Hussein Onn Malaysia (UTHM), Pagoh Campus. The working dataset was made by combining two primary sources. The first source is the official examination timetable for the first semester of the 2024/2025 academic year. Based on this, all examination sessions scheduled in the Multi-Purpose Hall (MPH) were extracted. This raw examination schedule contained all exams with their details such as course code, section number, course name, number of students enrolled, and the date and time originally scheduled.

The second source is the official list of courses that the university provides on each academic programme. This course list was aligned with the course code with the relevant academic programme (BWA, BWC, BWQ, BWS, BWW) and study year. One of the key processes was to merge and filter these two sources of information to obtain a clean dataset to analyse. All the exams in the schedule were verified with the course list. All other exams were dropped, only exams that are related to the five target programmes were left. The retained exams were then marked with their appropriate programme and year.

It is important to understand the course code format. An offering of a course is denoted by a certain code such as "BWJ205031_1". "BWJ20503" is the base course code. The section is denoted by the underscore and the number that follows the underscore. BWJ205031 and BWJ20503_2 were treated as individual exams due to different groups of students in these sections of the same course. Thus, in the analysis, the combination of unique courses and sections is considered a separate exam that should be scheduled. This precise modelling is essential to the correct identification of scheduling conflicts among student groups.

This preparation process is performed to ensure the dataset reflects the actual scheduling situation. After the data preparation phase, the final dataset was prepared to be analysed. It consisted of 34 distinct examinations and included 941 students in 5 academic programmes across four-year levels. This dataset provides a solid foundation for testing and comparing the two solution methods under realistic conditions.

Table 1 gives a portion of the resulting processed data.

Table 1 Part of processed data

Course Code & Section	Programme	Year	Students
BWA10303_1	BWA	1	22
BWA10303_2	BWQ	1	33
BWJ20503 _1	BWW	2	16
BWS10103_1	BWS	1	28
BWC30203_1	BWC	3	50

2.2 Conflict Matrix Construction

The core scheduling constraint is the avoidance of student clashes. A conflict exists between two examinations if they share students from the same programme and the same year of study [1]. To systematically encode all pairwise prohibitions, a symmetric binary conflict matrix C is constructed [19].

Let $E = \{e_1, e_2, \dots, e_n\}$ represent the set of n examinations, where $n = 34$. The conflict matrix C is an $n \times n$ matrix. Each element c_{ij} of this matrix is defined by the following rule:

$$c_{ij} = \begin{cases} 1 & \text{if examinations } e_i \text{ and } e_j \text{ share a programme} - \text{year cohort} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The matrix is symmetric ($c_{ij} = c_{ji}$) and has zeros on its diagonal ($c_{ii} = 0$).

The conflict density ρ, a measure of problem's constrainedness, is calculated as the ratio of actual conflict pairs to the total possible pairs [20]:

$$\rho = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij}}{\frac{n(n-1)}{2}} \times 100\% \tag{2}$$

This matrix C provides a complete specification of all hard constraints. The matrix representation can be easily converted into a conflict graph and vice versa [21].

In this study, conflicts are determined by shared programme-year student cohorts. In actual academic settings, there is usually an additional hard constraint where all sections of the same course must be examined simultaneously. In this model, this constraint was not applied to maintain the focus on the main comparative analysis of the two heuristic approaches in the framework of a common conflict.

An equivalent representation is an undirected graph $G = (V, E)$, where each vertex $v \in V$ corresponds to an examination and an edge $(v_i, v_j) \in E$ exists if and only if $c_{ij} = 1$ [21]. This graph-based representation is fundamental to the Graph Colouring approach and serves as input for Simulated Annealing [15].

2.3 Graph Colouring (GC) Model Implementation

Graph Colouring is a well-established heuristic for timetabling problems where the vertices represent events (such as exams) and the edges represent conflicts between them [11]. The approach maps the problem directly onto the classic vertex coloring problem from graph theory. Using the conflict graph $G = (V, E)$, the objective is to assign a color, representing a unique time slot, to each vertex such that adjacent vertices receive different colors. Adjacent nodes that have the same colour indicate a scheduling conflict [21].

The objective can be expressed mathematically as:

$$\text{Minimize } k \text{ subject to: } colour(v_i) \neq colour(v_j), \quad \forall (v_i, v_j) \in E \tag{3}$$

where k denotes the total number of colours used.

The Greedy Graph Colouring algorithm is adopted since it is simple, computationally efficient and effective in generating conflict-free timetables [22]. The algorithm is used to assign the colour to examinations in each order in a predefined order, such as by decreasing the degree of conflict. All the vertices (exams) have not been used by the adjacent nodes and have the colour which is chosen and the time slot is assigned to it. This process ensures that there is a conflict-free schedule. Although greedy colouring does not imply an optimal solution, it always results in feasible schedules with low computational cost, hence is widely used as a baseline in timetabling research [23][24].

2.4 Simulated Annealing (SA) Model Implementation

Simulated Annealing (SA) is a metaheuristic optimisation method that is inspired by a physical annealing process in the field of metallurgy, where a system is allowed to cool at a controlled temperature to a low-energy state [13]. To reproduce this process, the algorithm employs a probabilistic approach to accept both improving and worsening solutions in the initial stages of search to help the algorithm in escaping the local optima and to perform a more intensive search in the solution space [25]. SA has been widely studied and implemented in timetabling due to its simplicity and adaptability to domain-specific constraints [18].

2.4.1 Solution Representation and Cost Function

A candidate solution S is represented as a mapping that assigns each examination e_i to a time slot t_j. The quality of a solution is measured using a composite cost function based on the hard and soft constraints [26]:

$$C(S) = w_1 \cdot |T| + w_2 \cdot N_{conflicts} + w_3 \cdot N_{capacity} \tag{4}$$

where $|T|$ is the number of slots used, $N_{conflicts}$ and $N_{capacity}$ are penalty terms for student conflicts and capacity violations, and w_1, w_2, w_3 are weighting coefficients. A practical cost function must clearly reflect the influence of all constraint types to guide the SA search in a meaningful way. The weighting parameters are selected in such a way that $w_3, w_2 \gg w_1$, so that feasibility with the hard constraints is prioritised over timetable compactness [10].

2.4.2 SA Parameters and Cooling Schedule

The SA parameters are determined based on the standard recommendations and preliminary tuning [14][7]. The algorithm begins with an initial feasible solution and a high initial temperature T_0 = 100. A geometric cooling schedule controls the search intensity, defined by $T_{k+1} = \gamma \cdot T_k$ with a cooling rate $\gamma = 0.95$ [27]. The algorithm terminates when the temperature falls below a minimum threshold $T_{\min} = 0.1$ or after a maximum of 1000 iterations.

At each iteration, a neighbouring solution is generated by randomly selecting an examination and reassigning it to a different time slot [24]. The algorithm searches the solution space by generating a neighbouring solution S' from the current solution S. The change in the quality of the solution is computed as $\Delta C = C(S') - C(S)$. If $\Delta C \leq 0$, the new solution is accepted. If $\Delta C > 0$, the worse solution can still be accepted with a probability according to a Metropolis criterion:

$$P_{accept} = e^{-\Delta C / T} \tag{5}$$

where T is the current temperature [6]. This probabilistic acceptance mechanism allows the algorithm to escape local minima [26]. The temperature reduces gradually based on the cooling schedule, which guides the search toward a refined, high-quality final timetable.

2.5 Performance Evaluation Metrics

A comprehensive evaluation requires multiple performance criteria. The following metrics are used to compare the timetables generated by Graph Colouring and Simulated Annealing:

- Conflict Avoidance: Number of Student Conflicts
- Feasibility Compliance: Number of Capacity Violations.
- Schedule Compactness: Number of Time Slots Used, Examination Density (exams per slot).
- Load Distribution: Maximum Students per Slot, Minimum Students per Slot, Variance of Student Load.
- Computational Performance: Execution Time (in seconds).
- These metrics offer multi-dimensional assessment of the feasibility of each timetable, its efficiency, and balance, which will be the foundation of a comparative analysis.

3. Results and Discussion

In this section, the empirical findings of the application of GC and SA models to timetabling problem analysis are presented, followed by a comparative study. The comparison uses conflict resolution, timetable compactness, student load distribution, and computational efficiency as the metrics of evaluation of two approaches.

3.1 Conflict Analysis Results

The processed data consisted of 34 examinations in five academic programmes (BWA, BWQ, BWS, BWC, and BWW) and four-year levels, with a total of 941 students. To model examination clashes due to shared programme-year students' cohorts, a symmetric conflict matrix $C_{34\times34}$ was constructed. Overall, a total of 35 conflict pairs with a conflict density of 6.2% were identified. Table 1 summarises the distribution of conflicts.

Table 2 Conflict distribution by programme-year group

Programme	Year	Exams	Conflicts	Conflict Density
BWA	1	1	0	0%
BWA	2	3	3	100%
BWA	3	2	1	100%
BWC	1	3	3	100%
BWC	2	1	0	0%
BWC	3	1	0	0%
BWC	4	3	3	100%
BWQ	1	2	1	100%
BWQ	3	4	6	100%
BWQ	4	1	0	0%
BWS	1	1	0	0%
BWS	2	2	1	100%
BWW	2	2	1	100%
BWW	3	6	15	100%
BWW	4	2	1	100%

The internal conflicts were observed in ten programme-year groups with the highest concentration in BWW Year 3. This group included six examinations and fifteen conflict pairs. This conflict matrix is used as the primary input for both GC and SA, ensuring consistent constraint enforcement during timetable generation.

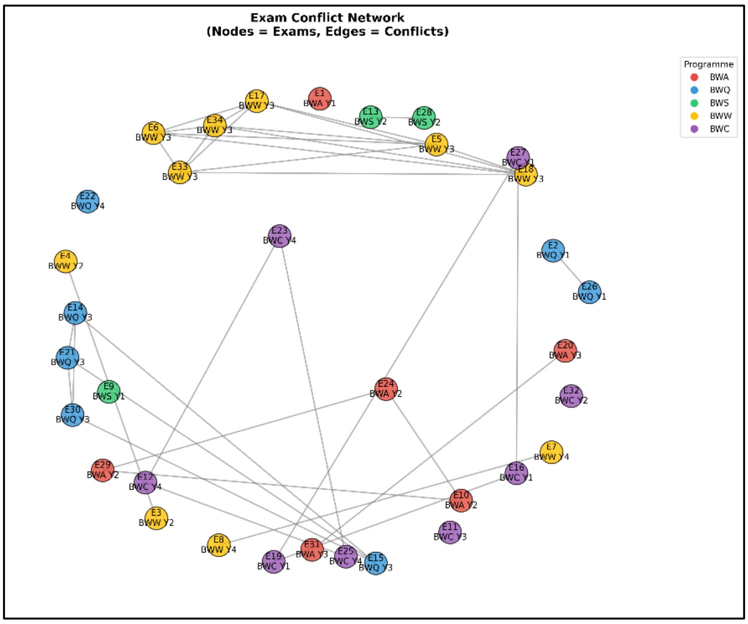

Fig. 1 Conflict Network Graph of Examination Dataset

Fig. 1 shows the conflict network graph, with nodes and edges representing exams and conflicts respectively. The definition of cohort-based conflict is reflected in the sparse but clustered structure. Both scheduling methods used this conflict matrix as the standard input to be compared fairly under the same set constraints.

3.2 Timetable Generation Results

Both algorithms successfully generated feasible timetables with no conflicts between students and no hall capacity violations, satisfying all defined hard constraints.

3.2.1 Graph Colouring Timetable Results

The Greedy Graph Colouring algorithm produced a feasible timetable that utilized 6 time slots, with an execution time of 0.0056 seconds. The examination timetable is divided into individual examination sessions, with each slot corresponding to a specific day and period, such as Monday Morning. Table 2 presents the distribution of student loads in each of these sessions.

Table 3 Student Load Distribution Across Time Slots for Graph Colouring

Slot	Session (Day & Period)	Exams	Students
1	Monday (Week 1) Morning	8	387
2	Monday (Week 1) Afternoon	7	286
3	Tuesday (Week 1) Morning	5	187
4	Tuesday (Week 1) Afternoon	6	41
5	Wednesday (Week 1) Morning	5	20
6	Wednesday (Week 1) Afternoon	3	20

All 34 examinations were scheduled within three unique days of a week, thus minimizing the exam period. Although achieving a high exam density of 5.67 exams per slot, indicating strong compactness of timetable, but it leads to weak load balancing. As a result, the distribution of students was highly uneven, with Slot 1 peaking at 387 students while Slots 5 and 6 had a minimum load of only 20 students. This shows the limitations of the algorithm in completely solving practical workload balancing problems. The GC-generated timetable is presented in Table 3. It demonstrates how multiple exams are packed into the same session.

Table 4 Graph Colouring Timetable

Session (Day & Period)	Exam ID	Session (Day & Period)	Exam ID
Monday (Week 1) Morning	BWA10303_1	Monday (Week 1) Afternoon	BWJ41303_2
Monday (Week 1) Morning	BWA10303_2	Monday (Week 1) Afternoon	BWB31203_1
Monday (Week 1) Morning	BWJ20503 _1	Monday (Week 1) Afternoon	BWC10303_1
Monday (Week 1) Morning	BWJ30403 _1	Monday (Week 1) Afternoon	BWC41203_1
Monday (Week 1) Morning	BWJ41303_1	Monday (Week 1) Afternoon	BWA21503_1
Monday (Week 1) Morning	BWS10103_1	Monday (Week 1) Afternoon	BWB10403_1
Monday (Week 1) Morning	BWA20303_1	Monday (Week 1) Afternoon	BWS20903_1
Monday (Week 1) Morning	BWC30203_1	Monday (Week 1) Afternoon	BWA32803_1
Monday (Week 1) Morning	BWC40203_1	Tuesday (Week 1) Morning	BWJ30103_1
Monday (Week 1) Morning	BWS21003_1	Tuesday (Week 1) Morning	BWB31002_1
Monday (Week 1) Morning	BWB44603_1	Tuesday (Week 1) Morning	BWC30503_1
Monday (Week 1) Morning	BWC10102_1	Tuesday (Week 1) Morning	BWC11402_1
Monday (Week 1) Morning	BWA31403_1	Tuesday (Week 1) Morning	BWA20403_1
Monday (Week 1) Morning	BWB44103_1	Tuesday (Week 1) Afternoon	BWJ30103_2
Monday (Week 1) Morning	BWC21003_1	Tuesday (Week 1) Afternoon	BWA31603_1
Monday (Week 1) Afternoon	BWJ20503 _2	Wednesday (Week 1) Morning	BWJ30503_1
Monday (Week 1) Afternoon	BWJ30403 _2	Wednesday (Week 1) Afternoon	BWJ30503_2

Enhanced Knowledge in Sciences and Technology Vol. 6 No. 1 (2026) 491-500

3.2.2 Simulated Annealing Timetable Results

The Simulated Annealing algorithm generated a conflict-free timetable with 16 time slots and an execution time of 0.4599 seconds. The final solution value of 196.58 gave the optimal solution cost, which reflects a balance between slot minimisation and load distribution. Table 4 indicated that the student per slot load was much more balanced.

Table 5 Student Load Distribution Across Time Slots for Simulated Annealing

Slot	Session (Day & Period)	Exams	Students	Slot	Session (Day & Period)	Exams	Students
1	Monday (Week 1) Morning	2	47	9	Friday (Week 1) Morning	3	94
2	Monday (Week 1) Afternoon	2	42	10	Friday (Week 1) Afternoon	1	50
3	Tuesday (Week 1) Morning	2	47	11	Monday (Week 2) Morning	3	91
4	Tuesday (Week 1) Afternoon	2	63	12	Monday (Week 2) Afternoon	1	20
5	Wednesday (Week 1) Morning	3	62	13	Tuesday (Week 2) Morning	2	43
6	Wednesday (Week 1) Afternoon	2	74	14	Tuesday (Week 2) Afternoon	3	76
7	Thursday (Week 1) Morning	2	74	15	Wednesday (Week 2) Morning	2	49
8	Thursday (Week 1) Afternoon	2	66	16	Wednesday (Week 2) Afternoon	2	43

A better load balance was achieved in the timetable generated using Simulated Annealing. Having exams spread over a period of 2 weeks and with 16 time slots, the maximum student load was 94 students in Slot 9, and the minimum load was 20 students in Slot 12. This represents a 75.7 percent decrease in peak load as compared to the Graph Colouring result. Table 5 presents the examination timetable developed by SA. The exams are distributed on two weeks and five different days.

Table 6 Simulated Annealing Timetable

Session (Day & Period)	Exam ID	Session (Day & Period)	Exam ID
Monday (Week 1) Morning	BWJ30403 _2	Friday (Week 1) Morning	BWJ20503 _2
Monday (Week 1) Morning	BWS20903_1	Friday (Week 1) Morning	BWA31403_1
Monday (Week 1) Afternoon	BWB10403_1	Friday (Week 1) Morning	BWC30503_1
Monday (Week 1) Afternoon	BWC11402_1	Friday (Week 1) Afternoon	BWC30203_1
Tuesday (Week 1) Morning	BWS21003_1	Monday (Week 2) Morning	BWJ41303_2
Tuesday (Week 1) Morning	BWJ30503_1	Monday (Week 2) Morning	BWC40203_1
Tuesday (Week 1) Afternoon	BWB31203_1	Monday (Week 2) Morning	BWA21503_1
Tuesday (Week 1) Afternoon	BWC10303_1	Monday (Week 2) Afternoon	BWJ30503_2
Wednesday (Week 1) Morning	BWS10103_1	Tuesday (Week 2) Morning	BWA10303_1
Wednesday (Week 1) Morning	BWB44603_1	Tuesday (Week 2) Morning	BWA31603_1
Wednesday (Week 1) Morning	BWC21003_1	Tuesday (Week 2) Afternoon	BWB44103_1
Wednesday (Week 1) Afternoon	BWJ30403 _1	Tuesday (Week 2) Afternoon	BWC41203_1
Wednesday (Week 1) Afternoon	BWA20303_1	Tuesday (Week 2) Afternoon	BWA32803_1
Thursday (Week 1) Morning	BWJ30103_1	Wednesday (Week 2) Morning	BWA10303_2
Thursday (Week 1) Morning	BWB31002_1	Wednesday (Week 2) Morning	BWJ20503 _1
Thursday (Week 1) Afternoon	BWC10102_1	Wednesday (Week 2) Afternoon	BWJ41303_1
Monday (Week 1) Afternoon	BWA20403_1	Wednesday (Week 2) Afternoon	BWJ30103_2

3.3 Comparative Performance

A direct comparison of the two timetables is presented in Table 6.

Table 7 Performance comparison of GC and SA timetables

Metric	Graph Colouring	Simulated Annealing
Examination Sessions Used	6	16
Days Used	3	5
Weeks Required	1	2
Max Students per Slot	387	94
Min Students per Slot	20	20
Student Load Variance	20239	366
Execution Time (Seconds)	0.0056	0.4599
Exam Density (Exams/Slot)	5.67	2.12
Conflicts Found	0	0
Capacity Violations	0	0

The results demonstrate a fundamental trade-off between scheduling compactness and operational balance. Graph Colouring is preferable when speed and scheduling compactness are prioritized. It is 82 times faster and uses far fewer time slots, allowing exams to be completed in one week. In contrast, Simulated Annealing is the better choice when creating a balanced and operationally feasible schedule is most important. It drastically reduces peak hall occupancy and spreads the examination load over more days.

3.4 Discussion of Trade-offs and Practical Implications

The comparative results show that the choice of algorithm should be guided by institutional priorities and resource constraints. There is no single best method, only the best method for a given situation.

Graph Colouring is the tool for speed and simplicity. It produces a full exam schedule almost instantly and fits all exams into the fewest possible days. This is ideal for small colleges, last-minute changes, or when the exam period must be kept as short as possible. If the main goal is to create a timetable as quickly as possible and to minimise the total number of examination days, then Graph Colouring is the recommended approach. This method is suitable for smaller institutions or situations where administrative time is very limited. The trade-off is that it can pack hundreds of students into one room at the same time, which demands a large hall and careful supervision.

Instead, Simulated Annealing is the tool for balance and manageability. It distributes students evenly during the exam period. This facilitates easy management of every session of examination, less crowding and a smooth examination session by many students. It involves higher cost in terms of exam time and creation time. This method is more effective when a large university is involved, or when a fair distribution of students is a primary consideration.

The decision is all dependent on the specific needs of the institution. Graph Colouring is the preferred method for speed and a compact schedule whereas Simulated Annealing is the preferred method of balance and simplified logistics.

4. Conclusion

In this study, a direct comparative analysis was made on two heuristic techniques namely Greedy Graph Colouring and Simulated Annealing, in solving a real-world Examination Timetabling Problem. The findings verify a high-performance trade-off that is influenced by design of algorithms.

Graph Colouring performed better in speed and compactness. It allocated 34 exams in only six time slots within milliseconds. Nonetheless, this efficiency was achieved at the cost of load balance, with up to 387 students in one session. Simulated Annealing, on the other hand, generated a more operationally practical timetable that minimizing peak load by more than 75% to 94 students per session, but needed much more time slots and computational time.

The selection of method does not depend on superiority, but rather on the factors that the institution prioritizes. The main scheduling objective of this study is to minimize the examination period. Both algorithms are aimed to reduce the number of time slots used, but Graph Colouring prioritizes more directly at this goal, thus achieving greater compactness. Therefore, the Graph Colouring method offers a more practical solution due to the critical importance of the compactness of the schedules. The possibility to fit all the examinations within one week and with minimum computing costs brings a physical administrative benefit to many institutions.

This study was limited to a single venue and failed to include invigilator or multi-room constraints. After that, the conflict model was only based on programme-year student cohorts and did not enforce the academic requirement that all sections of the same course should be examined at the same time. This comparison should be generalized to multi-venue scheduling in the future, more real-world constraints should be integrated, and a hybrid approach should be considered that targets the initial speed of GC with the refining balance of SA.

Acknowledgement

The authors would like to thank the Faculty of Applied Sciences and Technology at Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author Contribution

*The authors confirm their contribution to the paper as follows: **study conception and design**: Chang Chu Yun, Wan Nor Munirah Ariffin; **data collection:** Chang Chu Yun; **analysis and interpretation of results**: Chang Chu Yun, Wan Nor Munirah Ariffin; **draft manuscript preparation**: Chang Chu Yun, Wan Nor Munirah Ariffin. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling, 12(1)*, 55–89. https://doi.org/10.1007/s10951-008-0077-5

[2] Burke, E., Eckersley, A., McCollum, B., Petrovic, S., & Qu, R. (2010). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research, 206(1)*, 46–53. https://doi.org/10.1016/j.ejor.2010.01.044

[3] Schaerf, A. (1999). A Survey of Automated Timetabling. *Artificial Intelligence Review, 13*, 87–127. https://doi.org/10.1023/A:1006576209967

[4] Bashab, A., Osman Ibrahim, A., Abakar Tarigo Hashem, I., Aggarwal, K., Mukhlif, F., Ghaleb, F. A., & Abdelmaboud, A. (2023). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. *Computers, Materials & Continua, 74(3)*, 6461–6484. https://doi.org/10.32604/cmc.2023.034051

[5] Bardadym, V. A. (1995). Computer-aided school and university timetabling: The new wave. In *International Conference on the Practice and Theory of Automated Timetabling*. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 22-45. https://doi.org/10.1007/3-540-61794-9_50

[6] Ekanayake, E. M. M. B., Vithana, S. S. P., Ekanayake, E. M. H. E. B., Rathnayake, A. R. M. A. N., Abeysekara, A. M. R., Oorloff, T. S. J., & Senaratne, A. (2019). Mapping ilmenite deposit in Pulmudai, Sri Lanka using a hyperspectral imaging-based surface mineral mapping method. *Journal of the National Science Foundation of Sri Lanka, 47(3)*, 271. https://doi.org/10.4038/jnsfsr.v47i3.9276

[7] Nguyen, V. D., & Nguyen, T. (2021). An SHO-based approach to timetable scheduling: a case study. *Journal of Information and Telecommunication, 5(4)*, 421–439. https://doi.org/10.1080/24751839.2021.1935644

[8] Siew, E. S. K., Sze, S. N., Goh, S. L., Kendall, G., Sabar, N. R., & Abdullah, S. (2024). A survey of solution Methodologies for exam timetabling problems. *IEEE Access, 12*, 41479–41498. https://doi.org/10.1109/access.2024.3378054

[9] Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[10] Feng, X., Lee, Y., & Moon, I. (2017). An integer program and a hybrid genetic algorithm for the university timetabling problem. *Optimization Methods and Software, 32(3)*, 625-649. https://doi.org/10.1080/10556788.2016.1233970

[11] Aldeeb, B. A., Al-Betar, M. A., Abdelmajeed, A. O., Younes, M. J., AlKenani, M., Alomoush, W., ... & Alqahtani, M. A. (2019). A comprehensive review of Uncapacitated university examination. *International Journal of Applied Engineering Research, 14(24)*, 4524–4547.

[12] Oke, A., Wakili, A., & Olayiwola, B. (2023). Examination timetable scheduling using graph coloring for Faculty of Science. *British Journal of Multidisciplinary and Advance Studies: Education, Learning, Training & Development, 4(2)*, 117–143.

[13] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220(4598)*, 671-680. https://doi.org/10.1126/science.220.4598.671

[14] Dong, T., Qi, X., Zhang, Q., Li, W., & Xiong, L. (2019). Overview on vision-based 3d object recognition methods. In *Lecture notes in computer science*. Cham: Springer International Publishing. pp. 243–254. https://doi.org/10.1007/978-3-030-34110-7_21
[15] Ogunkan, S. K., Idowu, P. O., Omidiora, E. O., & Oyeleye, C. A. (2024). First fit Algorithm: A graph coloring approach to Conflict-Free university course timetabling. *Asian Journal of Research in Computer Science, 17(5)*, 125–139. https://doi.org/10.9734/ajrcos/2024/v17i5443
[16] Van Bulck, D., Goossens, D., & Schaerf, A. (2024). Multi-neighbourhood simulated annealing for the ITC-2007 capacitated examination timetabling problem. *Journal of Scheduling*. https://doi.org/10.1007/s10951-023-00799-1
[17] Battistutta, M., Schaerf, A., & Urli, T. (2017). Feature-based tuning of single-stage simulated annealing for examination timetabling. *Annals of Operations Research*, *252*(2), 239-254. https://doi.org/10.1007/s10479-015-2061-8
[18] Petrovic, S., & Burke, E. K. (2004). University timetabling. In J. Leung (Ed.), Handbook of Scheduling: *Algorithms, Models, and Performance Analysis* (pp. 45-1–45-23). Chapman & Hall/CRC.
[19] Cole, A. (1964). The preparation of examination time- tables using a small- store computer. *The Computer Journal, 7(2)*, 117–121. https://doi.org/10.1093/comjnl/7.2.117
[20] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications*. Journal of Operational Research Society, 47(3)*, 373-383. https://doi.org/10.1057/jors.1996.37
[21] Ghaffar, A., Din, I. U., Tariq, A., & Zafar, M. H. (2025). Hybridization and artificial intelligence in optimizing university examination timetabling problem: A systematic review. *Review of Education*, *13*(2), e70071. https://doi.org/10.1002/rev3.70071
[22] Aycan, E., & Ayav, T. (2009). Solving the course scheduling problem using simulated annealing. *IEEE International Advance Computing Conference, 8*, 462–466. https://doi.org/10.1109/iadcc.2009.4809055
[23] Bardadym, V. A. (1995). Computer-aided school and university timetabling: The new wave. In *International Conference on the Practice and Theory of Automated Timetabling*. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 22-45. https://doi.org/10.1007/3-540-61794-9_50
[24] Burke, E., Eckersley, A., McCollum, B., Petrovic, S., & Qu, R. (2010). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research, 206(1)*, 46–53. https://doi.org/10.1016/j.ejor.2010.01.044
[25] Oyeleye, Akinwale, C., Olabiyisi, Olatunde, S., Omidiora, Olusayo, E., Oladosu, & Babalola, J. (2012). Performance evaluation of simulated annealing and genetic algorithm in solving examination timetabling problem. *Scientific Research and Essays, 7(17)*. https://doi.org/10.5897/sre11.2103
[26] Burke, E., Eckersley, A., McCollum, B., Sanja, P., & Qu, R. (2003). Using simulated annealing to study behaviour of various exam timetabling data sets. In *The Fifth Metaheuristics International Conference 2003*. https://nottingham-repository.worktribe.com/output/1022344
[27] Abdelhalim, A., & Elkhayat, G. (2021). Enhanced simulated annealing with geometric cooling for examination timetabling. *International Journal of Advanced Computer Science and Applications, 12(6)*, 345–352. https://doi.org/10.3390/math9212734