MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 1, pp. 144–157 (2025)

Multi-neighborhood local search with room split balancer for exam timetabling: A case study

Siew E. S. K.[1], Sze S. N.[2], Goh S. L.[3], Hossin M.[2], Chiew K. L.[2]

[1] *Faculty of Computing and Software Engineering, i-CATS University College, 93350, Kuching, Sarawak, Malaysia*

[2] *Faculty of Computer Science & Information Technology, University Malaysia Sarawak, 94300, Kota Samarahan, Sarawak, Malaysia*

[3] *Optimisation and Visual Analytics Research Group, Faculty of Computing and Informatics, University Malaysia Sabah, Labuan International Campus, Labuan, 87000, Malaysia*

(Received 15 October 2024; Revised 10 February 2025; Accepted 15 February 2025)

This study explicitly addresses the examination timetabling problem (ETP) at University Malaysia Sarawak (UNIMAS), which encompasses both online and physical exams treated within a unified framework of uncapacitated and capacitated formulations. Currently, faculty exam timetabling managed by proprietary systems meets basic constraints but needs to incorporate faculty and stakeholder preferences into a mathematical formulation, making solution quality difficult to assess. To address this issue, we propose a mathematical model that includes university-wide constraints and considers extended soft constraints that accommodate faculty and stakeholder preferences for room sharing and achieving balanced exam splits for shared and non-shared exam scenarios. We introduce a two-stage multi-neighborhood local search method with a balancer to produce high-quality solutions that meet these constraints. Our approach outperforms existing proprietary systems by meeting all standard constraints and achieving extended soft constraints, improving scheduling efficiency and stakeholder satisfaction, and offering a more optimal solution for real-world exam timetabling.

Keywords: *exam timetabling; multi-neighborhood; local search; balance; timetabling; exam spreading.*

2010 MSC: 90-08 **DOI:** 10.23939/mmc2025.01.144

1. Introduction

The Exam Timetabling Problem (ETP) stands out as one of the complex scheduling issues due to its intricate nature and constraints. Recognized as a non-deterministic polynomial (NP) hard problem, ETP requires exponential computational time with an increasing problem size. ETPs are classified into incapacitated, where room capacities are disregarded, and capacitated, where room capacities are treated as hard constraints. This paper presents an ETP dataset derived from real-world scenarios at University Malaysia Sarawak (UNIMAS), encompassing online exams, treated as incapacitated formulations, and physical exams, treated as capacitated formulations, within a unified framework.

UNIMAS adopts a two-tiered approach to optimize exam scheduling. Firstly, the Pre-Graduate Studies Division centrally schedules generic subjects and elective courses; then, faculties independently oversee the scheduling of program-specific courses. Our study focuses on the latter. While some faculties schedule manually, others utilize a proprietary system, namely the FESS 2.0 system [1], which employs a two-stage heuristic: firstly, it clusters courses for concurrent examination, then allocates clusters to specific timeslots and rooms. To the best of the authors' understanding, a proper mathematical model for comprehending the underlying problem and assessing its optimality is lacking. Therefore, there is a need to formulate and develop a mathematical model, building upon the

This work was supported by a grant from Ministry of Higher Education of Malaysia.

constraints present. The manual timetabling is disregarded here due to its limited feasibility within desired timeslots and suboptimal room usage for large-scale exam scheduling.

Exam logistics present inherent challenges. One practical challenge involves sharing exams within rooms to minimize the number of rooms, a common practice among faculties. However, some faculties adamantly refuse to adopt this practice, adding a layer of complexity to the scheduling process. Furthermore, there is a frequent need to split rooms when specific exam sizes exceed the capacity of the largest available room, potentially leading to overcrowding or underutilization in certain rooms. For instance, consider a scenario where 205 students are assigned, with 100 in room A, 100 in room B, and only 5 in room C. Such disparities often confuse students in the minority room (C), potentially resulting in delays and disruptions during exams. Students are required to take the exam only in their allocated room. Therefore, achieving a balanced distribution of students across rooms for split exams is essential to ensure a smoother exam experience.

The contributions of this paper are three-fold: (i) we formulate the mathematical model for the UNIMAS ETP to understand the underlying problem constraints and assess its optimality comprehensively, (ii) include extended soft constraints on faculty preference in room sharing and balanced exam split, addressing the resistance from some faculties to adopt sharing practices; and (iii) introduce a multi-neighborhood local search approach to optimize scheduling while considering these extended soft constraints efficiently. We showcase applying the proposed approaches in real-world case studies on the Faculty of Economics and Business (FEB) and the Faculty of Cognitive Sciences and Human Development (FCSHD), demonstrating that the algorithm can effectively address these scheduling issues.

Section 2 covers related work, while Section 3 presents a mathematical formulation and its associated constraints with extensions. Section 4 provides a detailed description of our proposed algorithms. Experimental results are discussed in Section 5, and conclusions are presented in Section 6.

2. Related work

Prior research has documented surveys on the examination timetabling problem, covering formulations, solution methods, and algorithms, as evidenced by previous studies such as [2, 3]. In the survey presented in the latest study by Siew et al. [4], existing contemporary solution methodologies are categorized into several distinct categories: heuristics, mathematical optimization, matheuristics, metaheuristics, hybrid approaches, and hyper-heuristics. The widespread utilization of metaheuristics over the past decades has garnered significant attention and continues to grow in its application for solving optimization problems [5], including the ETP.

Metaheuristics, one of the most commonly used techniques in exam timetabling [4], operate iteratively, guiding and modifying subordinate heuristics to manipulate either a complete or partial single solution or a collection of solutions to generate high-quality solutions efficiently [6]. Subordinate heuristics may encompass a simple local search or a construction method. One significant advantage of metaheuristics lies in their nature as abstract search methods, allowing the application of their fundamental search logic to diverse problems characterized by basic components such as solution representation, quality assessment, and the concept of locality.

Locality refers to the practical generation of neighbouring solutions through heuristically guided functions based on either a single solution or a collection of incumbent solutions. During each iteration, local search relies on transitioning from the current solution to a neighbouring solution using a neighbourhood operator. This operator navigates the search space region surrounding the initial solution [7]. More recent contributions have predominantly utilized metaheuristic techniques, including local search methods, successfully applied to solve the ETP.

Müller [8] proposed a multi-phase algorithm, employing iterative forward search in the construction phase and integrating hill climbing and the great deluge. Bykov and Petrovic [9] extended hill climbing by incorporating a counting mechanism, demonstrating robust performance with the Saturation Degree heuristic. Burke and Bykov [10] introduced a late acceptance hill-climbing method, showcasing efficiency, and scale independence, particularly beneficial for larger instances. Bellio et al. [11] developed a two-stage simulated annealing approach utilizing multiple neighborhoods with proper parameter tuning, which achieved state-of-the-art results for the Toronto benchmark.

Recent studies have focused on the most renowned benchmarks, including the incapacitated Toronto benchmark [12], the capacitated Track 1 of the ITC 2007 benchmark [13], and research encompassing either or both of these benchmarks in [14–19]. Meanwhile, some studies address specific real-world cases by focusing on capacitated scenarios, as educational institutions' policies impose constraints and preferences, resulting in various iterations of the ETP. Notable examples of real-world case studies include University Malaysia Pahang in Malaysia [20], University Technology MARA [21], Purdue University in the USA [22], KU Leuven in Belgium [23], several Italian universities [24], and the Sepuluh Nopember Institute of Technology in Indonesia [25].

The capacitated formulation incorporates the clash-free constraint while encompassing period duration and room-related constraints. Specifically, it stipulates that each exam must occur in a separate room and that no exams should be split between rooms. Institutions in [26, 27] have policies that either prohibit splitting exams among multiple rooms or disallow simultaneous exams in the same room. Dammak et al. [28] permitted scheduling multiple exams within a single room but prohibited splitting exams across rooms.

In contrast, Komijan and Koupaei [29] devised a binary model for the ETP, allowing exams to be split into rooms on the same floor if the room capacity is exceeded, with room sharing permitted for a maximum of two exams per room. Genc and Sullivan [30] proposed a two-phase constraint programming model for the University College Cork ETP, allowing exams to be split across multiple rooms with one exception. Carlsson et al. [31] introduced a portfolio approach comprising both exact and metaheuristic methods, limiting exams to one per room but allowing for exams to be split across multiple rooms.

Laporte and Desroches [32] employed a room allocation subroutine wherein each room hosts a single exam. Larger exams are divided across multiple rooms, and efforts are made to distribute students evenly among rooms using a rooms balancer, but only in conditions where room sharing is not allowed. Our work seamlessly incorporates these principles. Moreover, we introduce a flexible framework that caters to shared and non-shared exams customized to faculty preferences. Within this framework, we maintain the imperative of balanced distribution for split exams, catering to the requirements of both shared and non-shared exam scenarios.

3. Problem formulation

Exam timetabling involves allocating resources such as students and rooms over limited periods to conduct pre-defined exams while adhering to hard and soft constraints [33]. Meeting hard constraints ensures a feasible timetable while minimizing violations of soft constraints. Table 1 presents a set of notations and variable definitions formally describing these constraints.

We utilize two binary variables as specified by:

$$x_{ik} = \begin{cases} 1, & \text{if exam } i \text{ is assigned to timeslot } k, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ir} = \begin{cases} 1, & \text{if exam } i \text{ is assigned to room } r, \\ 0, & \text{otherwise.} \end{cases}$$

The notations and variable definitions outlined in Table 1 indicate the hard constraint (section 3.1) and soft constraint (section 3.2) in the UNIMAS ETP, followed by the objective function (section 3.3). Finally, extended soft constraints (section 3.4) are given special consideration on a case-by-case basis.

3.1. Hard constraints

HC1: Each exam must be allocated to only one timeslot.

Table 1. Notations employed in for the problem formulation of ETP.

Parameter	Description
N	The number of examinations, $i, j \in \{1, 2, \ldots, N\}$, where $i \neq j$.
T	The number of timeslots, $k \in \{1, 2, \ldots, T\}$.
R	The number of rooms, $r, u \in \{1, 2, \ldots, R\}$, where $r \neq u$.
D	The total duration of the examination period (in days).
A	The number of distinct areas where rooms are located, $a, b \in \{1, 2, \ldots, A\}$, where $a \neq b$.
R_k	The number of rooms available at timeslot k.
v_r	The total capacity for room r.
u_r	The total unused capacity for room r.
m_i	The number of rooms where exam i has been split.
S	The number of students, $s \in \{1, 2, \ldots, S\}$.
S_i	The number of students enrolled in exam i.
c_{ij}	A conflict matrix, where each element c_{ij}, $i, j \in \{1, 2, \ldots, N\}$, represents the number of students who are simultaneously taking both exams i and j.

HC1 mandates that each exam be allocated to only one available timeslot, as specified in equation (1),

$$\sum_{k=1}^{T} \mathbf{x}_{ik} = 1. \tag{1}$$

HC2: No student is allowed to take two examinations simultaneously.

HC2 enforces a conflict-free requirement, wherein if examinations i and j are scheduled in timeslot k, the stipulation is that the number of students sitting for both examinations i and j must be 0, i.e., $c_{ij} = 0$. This stringent constraint is expressed in equation (2),

$$\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{T} \mathbf{x}_{ik} \mathbf{x}_{jk} \mathbf{c}_{ij} = 0. \tag{2}$$

HC3: The total number of students assigned to a room does not surpass its capacity.

HC3 ensures students assigned to a room do not exceed its capacity, as equation (3) indicates,

$$\mathbf{S}_i \leqslant \sum_{r=1}^{R} \mathbf{y}_{ik} \mathbf{v}_r. \tag{3}$$

HC4: The timeslots used must not exceed the duration of the planned exam session.

HC4 ensures that the number of time slots utilized does not exceed the number of days in the planned exam session, with two time slots allocated per day, as indicated in equation (4),

$$\frac{T}{2} \leqslant \mathbf{D}. \tag{4}$$

3.2. Soft constraints

SC1: Minimize room wastage by reducing the number of rooms used.

The cost of the count of rooms used in each period k, denoted by r_k, and summing these counts across all timeslots during the examination period in SC1, denoted as z_1, is illustrated in equation (5) along with equation (6),

$$\mathbf{z}_1 = \sum_{k=1}^{T} \mathbf{r}_k, \tag{5}$$

where

$$\sum_{i=1}^{N-1} \mathbf{x}_{ik} = \mathbf{r}_k \tag{6}$$

SC2: Spread each set of student examinations evenly.

The proximity cost in SC2, denoted as z_2, pertaining to the distribution of exams throughout the exam period, is depicted in equation (7). Equation (8) represents the weight of the penalty incurred when two exams, i and j, are scheduled with a gap of $|k_j - k_i|$ timeslots between them. The proximity values utilized in this context (16, 8, 4, 2, and 1) correspond to the formula's logic, where a larger gap between exams results in a decreased penalty. The proximity values introduced by Carter et al. [12] have been widely adopted in other research studies [20, 34, 35],

$$\mathbf{z}_2 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{c}_{ij} \mathbf{w}_{|k_j - k_i|}, \tag{7}$$

where

$$\mathbf{w}_{|k_j - k_i|} = \begin{cases} 2^{5-|k_j - k_i|}, & \text{if } 1 \leqslant |k_j - k_i| \leqslant 5, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

SC3: Minimize the number of rooms for a split examination; allocate rooms in the same area (building or campus) when possible.

The splitting cost in SC3, denoted as z_3, is illustrated in equation (9), while equation (10) defines m_i, representing the number of rooms to which exam i has been assigned. In cases where an exam is divided into multiple rooms, a penalty cost is assigned. Additionally, allocating an exam to different areas incurs a penalty of 1 for each additional area beyond the first. This penalty reflects the undesirability of splitting exams across multiple areas due to the increased complexity or inefficiency it introduces , which implies that for each additional area beyond the first, a penalty of 1 is added.

$$\mathbf{z}_3 = \sum_{i=1}^{N} \mathbf{m}_i + (\mathbf{A}_i - 1), \tag{9}$$

where

$$\mathbf{m}_i = \sum_{r=1}^{R} \mathbf{y}_{ir}. \tag{10}$$

SC4: Schedule exams within designated timeslots.

The undesired slot cost for exams not scheduled at designated timeslot or designated timeslot range in SC4, denoted as z_4, is illustrated in equation (11),

$$\mathbf{z}_4 = \sum_{i=1}^{N} (1 - \mathbf{x}_{i\bar{k}}). \tag{11}$$

3.3. Objective functions

According to the given mathematical expressions, the total soft conflict penalty cost for a solution can be formulated using equation (12). The objective is to minimize the total penalty, denoted as $z(X)$, in a feasible solution X. The penalty coefficients for the soft constraints, labelled as α_1, α_2, α_3, and α_4, are assigned values of 1, 1, 1, and 2, respectively. The penalty coefficient for α_4 carries a higher penalty compared to other constraints as it requires directly affects the faculty and structure of the exam schedule. Prioritizing adherence to designated timeslots restricts the flexibility to make improvements or adjustments for individual exams, limiting the options for rescheduling or reallocating resources. Consequently, this restriction amplifies the penalties associated with other soft constraints. In the context of general evaluation, the objective function will solely consider university-level soft constraints,

$$\mathbf{z}_X = \alpha_1 \cdot \mathbf{z}_1 + \alpha_2 \cdot \mathbf{z}_2 + \alpha_3 \cdot \mathbf{z}_3 + \alpha_4 \cdot \mathbf{z}_4. \tag{12}$$

3.4. Extended soft constraints

SE1: Flexible Exam Session – Faculties can shorten their exam sessions as long as they adhere to the overall duration allocated by the university for exams.

SE2: No Room Sharing – Certain faculties, like FCSHD, prefer not to share rooms for multiple exams unless those exams are combined and treated as a single entity.

SE3: Balance Room Allocation for Split Exam – Ensuring fair distribution of students across rooms for split exams.

4. Two-stage multi-neighborhood local search

Our solution approach comprises two stages: a constructive heuristic and hill climbing. In each stage, we utilize distinct neighborhood structures. In Stage I, we consider all hard constraints and slot preferences to find a feasible solution. In Stage II, we refine the initial solution to accommodate soft constraints, incorporating additional constraints derived from specific case studies. We explore feasible solutions throughout this process, ensuring conflict violations are avoided.

4.1. Neighborhood structure

In Stage I, we employed two neighbourhoods:

1. **Ejection move**, denoted by $EM(e,p)$, takes as attributes one exam $e \in E$ and one period $p \in P$. The move involves ejecting exam e from its period $p(e)$ and attempting to insert all exams from the Ejection Pool (EP) into $p(e)$, without causing any conflict violations. If at least one exam is inserted into $p(e)$, then e will be moved to the EP.
2. **Insertion move**, denoted as $IM(e,p)$, takes as attributes one exam $e \in E$ and one period $p \in P$. This move involves attempting to insert an exam e from the EP into a random period p, without causing any conflict violations.

In Stage II, six additional neighborhoods were introduced as follows, exploring only the feasible part of the search space.

1. **Shift move**, denoted by $SM(e,p,rl)$, takes as attribute one exam $e \in E$, and one period $p \in P$. The move involves assigning e in $p(e)$ and $rl(e)$, to a randomly selected period p and/or rooms rl, with $p(e) \neq p$. The move occurs if the size of the selected $rl \leqslant rl(e)$.
2. **Swap exams**, denoted by $SE(e_i,e_j)$, takes as attributes two exams e_i, $e_j \in E$, $e_i \neq e_j$, involves swapping e_i within period $p(e_i)$ and rooms $rl(e_i)$ with e_j within period $p(e_j)$ and rooms $rl(e_j)$. The move occurs if the size of the selected $rl \leqslant rl(e)$.
3. **Kick exam** [37], denoted by $KE(e_i,e_j,p)$, takes as attributes two exams e_i, $e_j \in E$, $e_i \neq e_j$ and one period $p \in P$. The move involves assigning $e_i \in p(e_i)$, to the period of e_j, denoted by $p(e_j)$, and assigning e_j to a randomly selected period p, with $p(e_i)$, $p(e_j) \neq p$. The move occurs if the size of the selected $rl \leqslant rl(e)$.
4. **Shift room**, denoted by $SR(e,rl)$, takes as attribute one exam $e \in E$, and one room list $rl \in R$. The move involves assigning exam e to a randomly selected room list rl, considering only rooms in the same area. The move is executed when the size of the $rl < rl(e)$, and $p(rl) \neq p(rl(e))$.
5. **Compact room within period**, denoted by $CR(e,rl)$, takes as attribute one exam $e \in E$, and one room list $rl \in R$. The move involves assigning e in to a randomly selected rl within the same period and non-empty room(s) in the same area. The move occurs if room sharing is allowed and the size of the selected $rl < rl(e)$ and $p(rl) = p(rl(e))$.
6. **Downsize room**, denoted by $DR(e,rl)$, takes as attribute one exam $e \in E$, and one room list $rl \in R$. The move involves assigning e in to a randomly selected rl within the same period and smaller sized empty room(s). The move occurs if sharing a room is prohibited and the size of the selected $rl < rl(e)$ and $p(rl) = p(rl(e))$.

In each iteration of Stage II, an united neighbourhood can be derived by combining the selected neighbourhoods with the random order to introduce an element of stochasticity and allow us to take full advantage of the search capabilities of each neighbourhood, considering that some of these neighbourhoods are complementary. These moves reject outcomes that violate hard constraints and adhere to specific soft constraint types, aiming to enhance elements related to other constraint violations.

They may improve one cost at the expense of another in the feasible part of the search space. If such moves lead to an overall deterioration in the objective, they are rejected.

For ease of reference, we symbolize the neighbourhoods associated with the eight moves above as NS_1, NS_2, NS_3, NS_4, NS_5, NS_6, NS_7, and NS_8 respectively. NS_1 and NS_2 are general basic neighbourhoods, while NS_6, NS_7, and NS_8 correspond to room-related moves, while NS_4 and NS_5 pertain to period-related moves. NS_3 is associated with both period and room move. Due to the simplification of constraints in NS_4 to NS_7, the search space of these neighbourhoods is slightly smaller than that of NS_3 and NS_8. Consequently, the search in neighbourhoods NS_4 to NS_7 is much faster than that in NS_3 and NS_8.

4.2. Constructive heuristic

We implement a two-step approach customized to suit the distinct requirements of both online and physical exams. Firstly, we schedule online exams into designated time slots, meticulously ensuring the absence of any conflicts. We assume that eLEAP can handle online exams without encountering any capacity constraints. Secondly, concerning physical exams, we prioritize exams with preferred time slots, ordered by large enrolment first. Subsequently, we allocate the remaining physical exams to available time slots, ensuring no scheduling conflicts and maximizing the efficient utilization of available rooms.

The room allocation process involves the following steps: initially, the algorithm utilizes the first-choice heuristic to identify single rooms capable of accommodating a given exam size. If none of the single rooms is adequate, the best-fit heuristic [38] is employed within the same area. In cases where splitting within the same area is not possible, we resort to the least preferable option, which involves dividing rooms across different areas. Any unscheduled exams are placed into the EP. If the EP is not empty at the end of the process, the algorithm initiates the utilization of neighborhoods NS_1 and NS_2 iteratively until all exams are scheduled, leaving no remaining exams in the EP. The initial feasible solution in this stage should meet all hard constraints HC1-HC4 and satisfy SC4.

4.3. HC-based multi-neighborhood local search

The Hill Climbing (HC) process involves taking non-worsening steps, allowing for repetitive application in a standard manner until a local minimum is attained. Similar to traditional greedy HC, a control parameter called the "cost threshold" is employed to accept candidate solutions with lower costs, while higher-cost solutions are rejected. A set of combined neighbourhoods in a random order in local search (CNLS) runs in each iteration. CNLS employs a single cost threshold for all neighbourhoods within one iteration, which is updated to the latest current solution at the end of the iteration to enhance the flexibility and efficiency of the search process, allowing for a thorough exploration of the solution space.

The process begins with an initial feasible solution generated in Stage I, with the cost threshold set to the initial solution's cost value. Throughout the search, if no shared room constraint is applied, then neighbors NS_3, NS_4, NS_5, NS_7, NS_8 will be selected. Otherwise, neighbors NS_3, NS_4, NS_5, NS_6 will be chosen. The algorithm exclusively accepts candidates with costs lower than the cost threshold or lower than the current solution cost. The cost threshold is then updated to the current cost at the end of each iteration, gradually decreasing and guiding the search towards a superior solution until the stopping criteria are met. Our CNLS approach is summarized in the algorithm depicted in Algorithm (1).

At the end of the algorithm, a Balancer is employed to distribute student allocations in rooms for split exams. The Balancer operates analogously to how an AVL tree [39] adjusts to maintain balance post-tree allocation, ensuring efficient searches. After optimizing timeslot and room allocations, this Balancer redistributes students across rooms allocated for split exams, maintaining a proportionate distribution relative to room capacity. Notably, no timeslot or room reallocation occurs during the balancing process. Consequently, adjustments to student numbers in split exams can be independently performed without incurring additional cost penalties.

Algorithm 1 HC-based Multi-Neighborhood Local Search Algorithm.

Require: $maxIteration, timeLimit$, a set of Exam (E), a set of neighbor(M);
initialization $current := initSolution$ // solution from Stage I
initialization $currentCost := calculate_{c}ost(current)$
for $i = 1, \ldots, maxIterationortime < timeLimit$
  $updateCostThreshold(currentCost)$
  $randomise(M)$ //Shuffle the list M
  for $j = 1, \ldots, size(M)$
    $(candidate, candidateCost) := generateNSolution(E, M)$ //generate a candidate solution
    if $candidateCost < currentCostorcandidateCost < thresholdCost$ **then**
      $current := candidate$
      $currentCost := candidateCost$
$best := current$
$balancer(best)$ //apply balancer function on the best solution

5. Experiment and result

5.1. Experiment case studies and settings

As of 2024, UNIMAS comprises ten faculties and offers over 40 undergraduate programs, accommodating 13 380 students. In this case study, our primary focus revolves around examination timetabling, utilizing datasets from two esteemed faculties: the Faculty of Economics and Business (FEB) and the Faculty of Cognitive Sciences and Human Development (FCSHD). These faculties, FEB and FCSHD, stand out due to their significant student populations and extensive course-student enrolments. Consequently, for specific exams, the sizes exceed the capacity of the largest available rooms, necessitating the split of rooms for exams.

In contrast to the case problem in existing literature, our examination timetabling addresses both online and physical exams, treating online exams as incapacitated and physical exams with capacitated considerations. In the FEB-01 dataset, there are 40 courses with a system graph displaying 196 edges and a 40x40 adjacency matrix, resulting in a density of 0.251. In the FEB-02 dataset, there are 47 courses with a system's graph exhibiting 237 edges and a 47x47 adjacency matrix, indicating a density of 0.219. In the FCSHD-02 dataset, there are 52 courses with a system's graph exhibiting 184 edges and a 52x52 adjacency matrix, representing a density of 0.144. Conflict density is calculated using the formula: (Number of conflicts)/(Total possible conflicts).

The characteristics of these datasets are outlined in Table 2. In the context of room resources, two types of rooms are available: small-sized faculty rooms, available in every timeslot, and medium to large-sized shared rooms, whose availability varies and is shared in a round-robin fashion among ten faculties. Room sizes are classified as small (<100), medium (100–150), and large (>150).

Table 2. The characteristics of datasets.

Instances	Exams	Students	Enrolments	Exam Type	Conflict Density	Room Size Distribution
FEB-01	40	1 921	6 026	Online & Physical	0.251	Small: 114, Medium: 24, Large: 10
FEB-02	47	2 131	7 151	Physical	0.219	Small: 91, Medium: 33, Large: 16
FCSHD-01	52	1 734	6 399	Physical	0.144	Small: 112, Medium: 25, Large: 14

The algorithm was implemented in Java and does not require any parameter setup. All experiments were conducted on a computer operating under Windows 11, equipped with an Intel(R) Core(TM) i7 CPU running at 2.80 GHz and 16GB of RAM. The CLNS algorithm underwent 30 independent runs for each dataset, halting after 15 iterations or within a 60-second time limit, whichever occurred first. The experiments in this study serve two primary purposes: first, to assess the effectiveness of the proposed approach in tackling the university objective function compared to the outcomes produced by the proprietary FESS, and second, to evaluate the proposed methods for enforcing extended soft constraints and ascertain the existence of feasible solutions.

5.2. Experiment result – university standard objective

The comparison between our results and those from FESS based on the standard objective is presented in Table 3. To ensure a fair comparison between our approach and the proprietary system, we standardize the time slots to align with those generated by the FESS system. Through experiments aimed at achieving standard objectives, superior results were obtained. Table 3 shows that CNLS consistently achieves lower mean values across all soft constraints (SC1 to SC4) and the overall total (Total Mean) compared to FESS. These results highlight CNLS's effectiveness in optimizing performance metrics across different constraints and datasets, outperforming FESS regarding average outcomes. However, this performance advantage comes with a trade-off in cost breakdown. Excelling in meeting specific soft constraints may lead to higher expenses in other areas, thereby achieving superior performance in certain aspects.

Table 3. The experimental result of CNLS algorithm.

Instances	**Approach**	**Soft constraints**									
		SC1		**SC2**		**SC3**		**SC4**		**Total**	
		Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
FEB-01	CNLS	59.0	61.0	21.3	23.2	**31.0**	31.6	**6.0**	6.0	**118.5**	121.8
	FESS	**58.0**	–	**18.0**	–	40.0	–	16.0	–	132.0	–
FEB-02	CNLS	**57.0**	59.1	**17.3**	19.5	**23.0**	24.7	4.0	4.1	**104.3**	107.3
	FESS	74.0	–	21.2	–	45.0	–	**2.0**	–	142.2	–
FCSHD-01	CNLS	**72.0**	75.2	**12.7**	16.1	**22.0**	25.6	6.0	6.0	**114.5**	123.0
	FESS	78.0	–	19.8	–	32.0	–	**4.0**	–	133.8	–

5.3. Experiment result – extended soft constraints

5.3.1. Reduced exam session

Besides fulfilling the hard constraints and the objective function, both faculties aim to satisfy the extended soft constraint SE1, the Flexible Exam Session constraint. Both faculties desire to tailor their schedules to shorten their exam sessions' duration rather than strictly adhering to the university's allocated timeframe. This adjustment aims to facilitate an earlier completion of exams. However, due to room constraints, they need more rooms within the desired shortened exam session.

The experiment involved applying the Stage I constructive heuristic to three datasets and the Toronto benchmark to assess the effectiveness of our approach in achieving the SE1 constraint, which consists of scheduling exams within minimum timeslots. Table 4 illustrates the timeslots between the FESS system/given timeslot limit and our constructive heuristic approach for the Toronto benchmark and the studied datasets. The initial feasible solutions for all the Toronto instances could fit into the given minimum timeslots. Furthermore, the timeslots could be reduced further for the studied datasets. Our approach has demonstrated its capability to achieve SE1 and minimize the overall exam session for each dataset. The * indicates the minimum number of timeslots for scheduling in the respective instances. However, it is essential to note that this reduction comes at the expense of a higher overall cost in the datasets.

5.3.2. Room sharing

Notably, FCSHD imposes an additional constraint, SE2, known as the No Room Sharing constraint. Unlike other faculties, which generally minimize the number of rooms used by sharing exams in one room, this constraint reflects FCSHD's preference against sharing rooms for multiple exams unless those exams are combined and treated as a single entity. Thus, all exams with shared rooms are adjusted

Table 4. The experimental result of CNLS algorithm.

Instances	FESS System/ Limited	Constructive Heuristic	
	Timeslots	Timeslots	Runtime (s)
FEB-01	10	10	0.20
FEB-02	14	12*	0.22
FCSHD-01	14	12*	1.66
CAR91	35	35	11.16
CAR92	32	32	46.53
EAR83	24	24	0.08
HEC92	18	18	40.73
KFU93	20	20	22.82
LSE91	18	18	1.85
RYE92	23	23	5.16
STA83	13	13	0.63
TRE92	23	23	1.51
UTA92	35	35	7.91
UTE92	10	10	1.51
YOR83	21	21	1.20

to remove any shared exams. The experiment involved applying the CLNS algorithm together with and to the FCSHD-01 dataset. As depicted in Figure 1, the box plots comparing the distribution of overall costs associated with shared and non-shared rooms demonstrate the significant impact of this constraint on scheduling solutions' costs. Solutions adhering to the SE2 constraint, which excludes shared rooms, demonstrate higher median costs than those utilizing shared rooms.

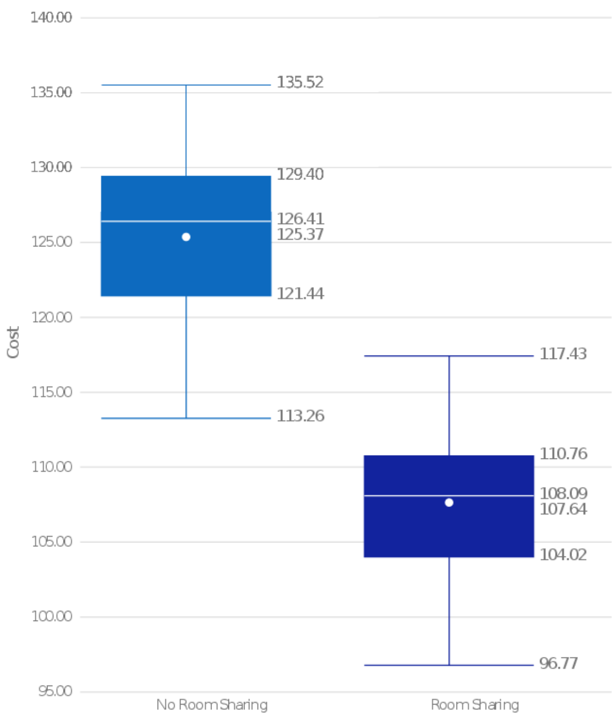

Fig. 1. Cost for shared and non-shared rooms for FCSHD-01.

5.3.3. Balance distribution

On the other hand, the SE3 constraint, the Balance Room Allocation for Split Exam constraint, ensures equitable distribution of students across rooms for split exams to prevent overcrowding or underutilization in any room. Currently, the splitting method employs a sequential occupancy approach. For example, if 205 students are allocated across rooms A, B, and C, the distribution would be as follows: Rooms A and B, each accommodating 100 students, are fully occupied, leaving room C with only

5 students. The minority of students in room C often encounter difficulty locating the correct room, leading to confusion and delays in the exam process. Consequently, students who arrive at the wrong room must be redirected to the correct one, resulting in wasted time and heightened stress levels. To address this issue, the exam administrator desires a balanced distribution to streamline the exam process.

In the context of where exams may be split across multiple rooms, it fulfills the fair distribution number of students based on the capacity of rooms for a split exam. A commonly used fairness measure in resource allocation called Jain's Fairness Index [40] is used. Let usedCap_{ir} denotes the used capacity and totalCap_{ir} denotes the total capacity of exam i in room r. The throughput of an exam, denoted as TP_i, is defined in equation (13). The Jain's Fairness Index $\text{JFI}(R)$ over n_e split exams, where n_e refers to the total number of split exams, is defined in equation (14),

$$\text{TP}_i = \frac{\sum_{r=1}^{M} \text{usedCap}_{ir}}{\text{totalCap}_{ir}}, \tag{13}$$

$$\text{JFI}(R) = \frac{\left(\sum_{i=1}^{n_e} \text{TP}_i\right)^2}{n_e \cdot \sum_{i=1}^{n_e} (\text{TP}_i)^2}. \tag{14}$$

The $\text{JFI}(R)$ falls within the range of $(0, 1]$, and its interpretation is intuitive. Specifically, when a solution results in a $\text{JFI}(R)$ equal to 1, it signifies that this solution represents a completely fair allocation. The objective is to enhance fairness by maximizing $\text{JFI}(R)$, indicating a more equitable distribution of resources. Any exams that are not split are omitted from the calculation. Additionally, the balancing function will not affect the objective function or the cost, as the balancing will be performed within the existing room allocation.

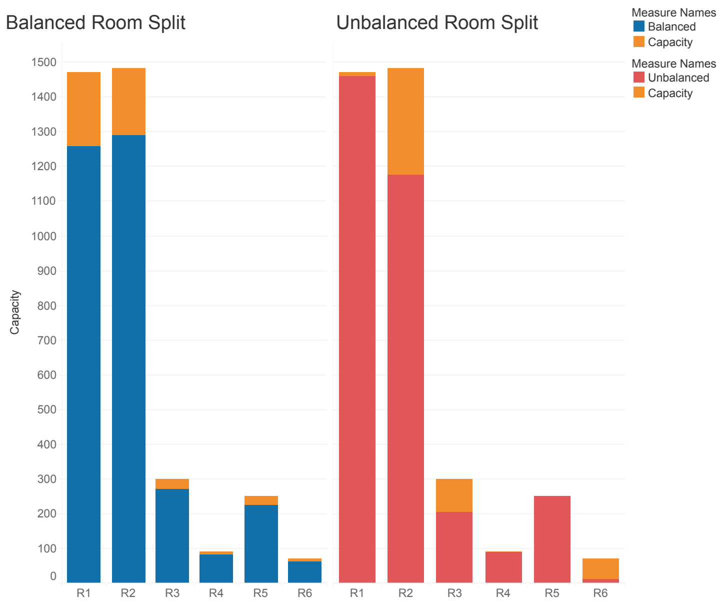

Fig. 2. Split room capacity usage: balanced vs. unbalanced.

The experiment involved applying the balancer function at the end of the algorithm to all datasets, addressing the balancing of split exams for both shared room and non-shared room scenarios. The results from Table 5 illustrate an improvement in the $\text{JFI}(R)$ values after the balancing of student distribution across multiple rooms for split exams. Across all instances, there is a consistent increase in $\text{JFI}(R)$ after balancing. Specifically, the percentage differences range from 3% to 13%. Additionally, Figure 2 visually represents the distribution of capacity usage relative to room capacity for balanced and unbalanced solutions. As depicted in the bar chart, the balanced solution exhibits a more even distribution of capacity usage across rooms than the unbalanced one, underscoring the positive impact of balancing measures on achieving fairness in student distribution during split exams.

Table 5. Jain's fairness index comparison.

Instances	JFI Before Balancing	JFI After Balancing	%Difference
FEB-01	0.894	0.923	3%
FEB-02	0.892	0.921	3%
FCSHD-01	0.881	0.999	13%

6. Conclusions

This study addresses the real-world exam timetabling challenges at UNIMAS, encompassing incapacitated and capacitated formulations within a unified framework. We formulated a mathematical model and devised a two-stage multi-neighborhood algorithm to tackle this complex problem. Our approach utilizes multi-neighborhood local search, which involves exploring a larger search space by balancing exploration across different neighborhoods and exploiting promising solutions within each. An equiroom balancer ensures an even distribution of students across rooms, accommodating both shared and non-shared exam scenarios. The results demonstrate that our proposed method outperforms the proprietary system in achieving standard objectives. Additionally, we conducted evaluations of our approach against extended soft constraints on a case-by-case basis, including shortening exam sessions, addressing room-sharing issues, and balancing student distributions for exam room allocation. In future research endeavors, we aim to incorporate datasets from other faculties at UNIMAS to account for the diverse aspects specific to each faculty. Moreover, our focus will expand toward achieving more efficient resource allocation at the university level rather than solely at the faculty level.

[1] Sze S. N., Phang M. H., Chiew K. L. Real-Life Faculty Examination Timetabling to Utilise Room Used. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). **9** (3–11), 51–54 (2017).

[2] Burke E. K., Petrovic S. Recent research directions in automated timetabling. European Journal of Operational Research. **140** (2), 266–280 (2002).

[3] Qu R., Burke E. K., McCollum B., Merlot L. T. G., Lee S. Y. A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling. **12**, 55–89 (2009).

[4] Siew E. S. K., Goh S. L., Kendall G., Sabar N. R., Abdullah S. A Survey of Solution Methodologies for Exam Timetabling Problems. IEEE Access. **12**, 41479–41498 (2024).

[5] Ma Z., Wu G., Suganthan P. N., Song A., Luo Q. Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms. Swarm and Evolutionary Computation. **77**, 101248 (2023).

[6] Swan J., Adriaensen S., Brownlee A. E. I., Hammond K., Johnson C. G., Kheiri A., Krawiec F., et al. Metaheuristics "in the large". European Journal of Operational Research. **297** (2), 393–406 (2022).

[7] Bianchi L., Dorigo M., Gambardella L. M., Gutjahr W. J. A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing. **8**, 239–287 (2009).

[8] Müller T. Real-life examination timetabling. Journal of Scheduling. **19**, 257–270 (2016).

[9] Bykov Y., Petrovic S. A step counting hill climbing algorithm applied to university examination timetabling. Journal of Scheduling. **19**, 479–492 (2016).

[10] Burke E. K., Bykov Y. The late acceptance hill-climbing heuristic. European Journal of Operational Research. **258** (1), 70–78 (2017).

[11] Bellio R., Ceschia S., Di Gaspero L., Schaerf A. Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling. Computers & Operations Research. **132**, 105300 (2021).

[12] Carter M. W., Laporte G., Lee S. Y. Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society. **47**, 373–383 (1996).

[13] McCollum B., McMullan P., Burke E. K., Parkes A. J., Qu R. The second international timetabling competition: Examination timetabling track. Technical Report QUB/IEEE/Tech/ITC2007/-Exam/v4.0/17, Queen's University, Belfast (2007).

[14] Van Bulck D., Goossens D., Schaerf A. Multi-neighbourhood simulated annealing for the ITC-2007 capacitated examination timetabling problem. Journal of Scheduling. 1–16 (2023).

[15] Alefragis P., Gogos C., Valouxis C., Housos E. A multiple metaheuristic variable neighborhood search framework for the Uncapacitated Examination Timetabling Problem. Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT. **1**, 159–171 (2021).

[16] Aldeeb B. A., Al-Betar M. A., Norwawi N. M., Alissa K. A., Alsmadi M. K., Hazaymeh A. A., Alzaqebah M. Hybrid intelligent water Drops algorithm for examination timetabling problem. Journal of King Saud University-Computer and Information Sciences. **34** (8), 4847–4859 (2022).

[17] Nand R., Reddy E., Chaudhary K., Sharma B. Preference Based Stepping ahead Firefly Algorithm for solving Real-World uncapacitated Examination Timetabling Problem. IEEE Access (2024).

[18] Mandal A. K., Kahar M. N. M., Kendall G. Addressing examination timetabling problem using a partial exams approach in constructive and improvement. Computation. **8** (2), 46 (2020).

[19] Leite N., Melício F., Rosa A. C. A fast threshold acceptance algorithm for the examination timetabling problem. Handbook of operations research and management science in higher education. 323–363 (2021).

[20] Kahar M. N. M., Kendall G. The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. European Journal of Operational Research. **207** (2), 557–565 (2010).

[21] Saat E. H. M., Ilham N. I., Othman N., Bakar Z. A., Yusof Y., Rahman N. H. A. The examination timetabling problem based on expert system: a case study in Malaysia. 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC). 121–126 (2019).

[22] Müller T. Real-life examination timetabling. Journal of Scheduling. **19**, 257–270 (2016).

[23] Woumans G., De Boeck L., Beliën J., Creemers S. A column generation approach for solving the examination-timetabling problem. European Journal of Operational Research. **253** (1), 178–194 (2016).

[24] Battistutta M., Ceschia S., De Cesco F., Di Gaspero L., Schaerf A., Topan E. Local search and constraint programming for a real-world examination timetabling problem. International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. 69–81 (2020).

[25] Muklason A., Pratama E. J., Premananda I. G. A. Generic University Examination Timetabling System with Steepest-Ascent Hill Climbing Hyper-heuristic Algorithm. Procedia Computer Science. **234**, 584–591 (2024).

[26] Merlot L. T. G., Boland N., Hughes B. D., Stuckey P. J. A hybrid algorithm for the examination timetabling problem. Practice and Theory of Automated Timetabling IV: 4th International Conference, PATAT 2002. Selected Revised Papers. 207–231 (2002).

[27] McCollum B., McMullan P., Parkes A. J., Burke E. K., Qu R. A new model for automated examination timetabling. Annals of Operations Research. **194**, 291–315 (2012).

[28] Dammak A., Elloumi A., Kamoun H. Classroom assignment for exam timetabling. Advances in Engineering Software. **37** (10), 659–666 (2006).

[29] Komijan A. R., Koupaei M. N. A new binary model for university examination timetabling: a case study. Journal of Industrial Engineering International. **8** (1), 28 (2012).

[30] Genc B., O'Sullivan B. A two-phase constraint programming model for examination timetabling at university college cork. Principles and Practice of Constraint Programming: 26th International Conference, CP 2020. **26**, 724–742 (2020).

[31] Carlsson M., Ceschia S., Di Gaspero L., Mikkelsen R. Ø., Schaerf A., Stidsen T. J. R. Exact and metaheuristic methods for a real-world examination timetabling problem. Journal of Scheduling. **26** (4), 353–367 (2023).

[32] Laporte G., Desroches S. Examination timetabling by computer. Computers & Operations Research. **11** (4), 351–360 (1984).

[33] McCollum B., Schaerf A., Paechter B., McMullan P., Lewis R., Parkes A. J., Di Gaspero L., Qu R., Burke E. K. Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS Journal on Computing. **22** (1), 120–130 (2010).

[34] Ayob M., Abdullah S., Malik A. M. A. A practical examination timetabling problem at the Universiti Kebangsaan Malaysia. International Journal of Computer Science and Network Security. **7** (9), 198–204 (2007).

[35] Burke E. K., Kingston J., De Werra D. Applications to timetabling. Handbook of graph theory. **445**, 4 (2004).

[36] Ding J., Shen L., Lü Z., Peng B. Parallel machine scheduling with completion-time-based criteria and sequence-dependent deterioration. Computers & Operations Research. **103**, 35–45 (2019).

[37] Di Gaspero L. Recolour, shake and kick: A recipe for the examination timetabling problem. Proceedings of the fourth international conference on the practice and theory of automated timetabling. 404–407 (2002).

[38] Siew S. K. E., Nah Sze S., Goh K. L., Wong J. J. Grouping and Heuristics for A Multi-stage Class Timetabling System. 2022 International Conference on Digital Transformation and Intelligence (ICDI). 111–116 (2022).

[39] Nurmi O., Soisalon-Soininen E., Wood D. Concurrency control in database structures with relaxed balance. Proceedings of the sixth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. 170 (1987).

[40] Jain R. K., Chiu D. M. W., Hawe W. R. A quantitative measure of fairness and discrimination. Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA. **21**, 1 (1984).

Багатосусідський локальний пошук із балансуванням розподілу кімнат для розкладу іспитів: практичний приклад

Сью Е. С. К.[1], Сзе С. Н.[2], Го С. Л.[3], Хоссін М.[2], Чью К. Л.[2]

[1] *Факультет обчислювальної техніки та розробки програмного забезпечення, Університетський коледж i-CATS, 93350, Кучінг, Саравак, Малайзія*

[2] *Факультет комп'ютерних наук та інформаційних технологій, Університет Малайзії Саравак, 94300, Кота Самарахан, Саравак, Малайзія*

[3] *Дослідницька група оптимізації та візуальної аналітики, факультет обчислювальної техніки та інформатики, Університет Малайзії Сабах, Міжнародний кампус Лабуан, Лабуан, 87000, Малайзія*

У цьому дослідженні явно розглядається проблема розкладу іспитів (ETP) в Університеті Малайзії Саравак (UNIMAS), який охоплює як онлайн, так і фізичні іспити, що розглядаються в єдиній структурі некомпетентних і компетентних формулювань. Наразі розклад іспитів викладачів, який керується власними системами, відповідає основним обмеженням, але потребує включення вподобань викладачів та зацікавлених сторін у математичне формулювання, що ускладнює оцінку якості рішення. Щоб розв'язати цю проблему, пропонується математична модель, яка включає загальноуніверситетські обмеження та враховує розширені м'які обмеження, які враховують уподобання викладачів і зацікавлених сторін щодо спільного використання кімнат і досягнення збалансованого розподілу іспитів для спільних та індивідуальних сценаріїв іспиту. Запроваджено двоетапний багатосусідський метод локального пошуку з балансуванням для створення високоякісних рішень, які відповідають цим обмеженням. Запропонований підхід перевершує існуючі запатентовані системи, дотримуючись усіх стандартних обмежень і задовольняючи розширені м'які обмеження, покращуючи ефективність планування та задоволеність зацікавлених сторін, а також пропонуючи більш оптимальне рішення для реального розкладу іспитів.

Ключові слова: *розклад іспитів; багатосусідство; локальний пошук; баланс; складання розкладу; поширення іспиту.*