

# AN INTEGER LINEAR PROGRAMMING MODEL FOR MAXIMIZING PREFERENCE SATISFACTION IN UNIVERSITY TIMETABLING

Pitchaya Wiratchotisation<sup>1</sup>, Nattapon Surasin<sup>1</sup>, Yanin Sareeso<sup>1</sup> and Sarun Paisarnsrisomsuk<sup>2,\*</sup>

*Received: December 31, 2025; Revised: February 20, 2026; Accepted: March 11, 2026*

## Abstract

Manual university course timetabling is a time-consuming, complex, and conflict-prone combinatorial optimization problem, widely recognized as NP-hard due to its intricate constraints. This research automates and optimizes this task for the Department of Statistics at our university using a novel Integer Linear Programming (ILP) model. Implemented in Python with the Gurobi solver, the model maximizes weighted instructor preferences while strictly adhering to hard constraints, including student cohorts, faculty availability, and room capacity, alongside specific pedagogical constraints necessitating consecutive periods for intensive subjects and non-consecutive days to optimize student study intervals. The model successfully scheduled all 27 courses to global optimality within 285.15 seconds, achieving a 0% MIP gap and an average preference satisfaction rate of 89.6%. The resulting system generates complete, conflict-free timetables that respect constraints such as lunch breaks and workload limits, demonstrating a significant improvement in efficiency and solution quality over traditional manual methods, offering a scalable solution applicable to broader academic contexts.

**Keywords:** Course Timetabling; Integer Linear Programming; Allocation Optimization; University Scheduling; Gurobi

## Introduction

Academic institutions worldwide face the recurrent and demanding task of the University Course Timetabling Problem (UCTTP). Fundamentally, this process involves allocating a set of events, such as lectures and labs, and resources including faculty

members, student groups, and specific room types to a finite number of available timeslots, subject to a diverse set of institutional rules and constraints. While the administrative goal is straightforward that is to produce a conflict-free schedule, the

---

<sup>1</sup> Department of Statistics, Faculty of Science, Khon Kaen University, Khon Kaen, 40002, Thailand., E-mail: [pitcwi@kku.ac.th](mailto:pitcwi@kku.ac.th), ORCID: 0000-0002-7482-1229; [nattapon.su@kkumail.com](mailto:nattapon.su@kkumail.com); [yanin\\_sa@kkumail.com](mailto:yanin_sa@kkumail.com)

<sup>2</sup> Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen, 40002, Thailand, E-mail: [sarunpa@kku.ac.th](mailto:sarunpa@kku.ac.th), ORCID: 0009-0002-4395-0834

\* Corresponding author

underlying mathematical reality is a combinatorial optimization problem of significant difficulty. The literature universally classifies UCTTP as NP-hard (Cooper & Kingston, 1995), a designation implying that no known algorithm can solve the problem in polynomial time. As the number of students, courses, and rooms increases, the computational time required to locate a feasible solution grows exponentially, creating a massive search space that manual methods often fail to navigate effectively.

The difficulty of the UCTTP stems primarily from the interaction between two distinct types of restrictions. The first category, known as hard constraints, consists of mandatory rules that determine the feasibility of a schedule. Examples include the physical capacity of a classroom or the requirement that a professor cannot teach two distinct courses simultaneously. Violation of a hard constraint renders a timetable invalid. The second category, soft constraints, dictates the quality of the schedule rather than its validity. These include preferences such as minimizing gaps in student schedules or accommodating faculty time requests (Burke & Petrovic, 2002). In real-world scenarios, these constraints often conflict, forcing administrators to make difficult trade-offs that rarely achieve optimal satisfaction for all parties.

Researchers have addressed these challenges through various methodologies that generally fall into two categories: heuristic approaches and exact methods. Metaheuristics, such as Genetic Algorithms (GA), Tabu Search (TS), and Simulated Annealing (SA), are frequently employed for large-scale, university-wide scheduling. GA mimics natural selection to evolve populations of schedules (Burke *et al.*, 1995), while TS (Alvarez-Valdes *et al.*, 2002) and SA (Elmohamed *et al.*, 1997) utilize local search mechanisms to escape local optima. These methods are highly scalable and capable of producing “good enough” solutions for massive datasets in short timeframes. However, they lack a guarantee of optimality; there is no mathematical certainty that the generated schedule is the best possible configuration. Alternatively, Constraint Programming offers flexibility in modeling complex logic but often struggles with pure optimization objectives compared to mathematical programming (Rossi-Doria *et al.*, 2003).

In contrast, exact methods, particularly Integer Linear Programming (ILP), treat timetabling as a system of linear equations where decision variables are restricted to integer values. For instance, a binary variable  $x_{crt}$  indicates whether course  $c$  is assigned to room  $r$  at time  $t$ . Solved using algorithms like Branch and Bound, ILP provides a rigorous advantage. It guarantees finding the

global optimal solution or definitively proving that no solution exists (Land & Doig, 2009). The historical trade-off has been scalability, as ILP can become computationally intractable for massive instances. However, for departmental-level scheduling where the problem size is finite and manageable, ILP offers superior results by strictly enforcing hard constraints and maximizing soft constraint satisfaction without the uncertainty inherent in heuristics.

Recent studies have demonstrated renewed interest in exact ILP methods for university timetabling, particularly at the departmental level where problem sizes remain tractable. Mallari *et al.* (2023) developed a multi-objective MILP model for coursework timetabling that achieved 9.45% improvement over manually constructed schedules while simultaneously optimizing curriculum compliance, student workload, and instructor workload. The International Timetabling Competition 2019, documented by Müller *et al.* (2025), established comprehensive benchmarks from ten universities worldwide, revealing that even large-scale real-world instances can be addressed through sophisticated exact and hybrid methods. At the departmental level, Mokhtari *et al.* (2021) applied multi-objective mixed-integer programming to postgraduate course scheduling at Islamic Azad University, demonstrating the viability of exact methods when the course set remains bounded. Similarly, Talmor (2024) reported that an ILP model successfully scheduled 236 classes with a 95% reduction in planning time compared to manual approaches. Harrabi *et al.* (2024) developed an ILP-based decision support system that prioritizes teacher preferences, achieving significant performance improvements over manual processes.

Our work extends existing research by developing an ILP model specifically tailored for the Department of Statistics at Khon Kaen University. Unlike university-wide problems that require the scalability of metaheuristics, our departmental scope with approximately 30 courses allows us to leverage the global optimality guarantee of ILP while incorporating novel pedagogical constraints. Specifically, we contribute: 1) A consecutive-period tracking mechanism ensuring multi-hour courses remain in contiguous blocks; 2) A non-consecutive-day constraint guaranteeing at least one day between weekly sessions of the same course; 3) Empirical validation showing 89.6% preference satisfaction with guaranteed feasibility in under 5 minutes. By embedding these complex rules directly into linear constraints, we demonstrate that exact methods can successfully accommodate nuanced academic

requirements while guaranteeing mathematical optimality.

## Materials and Methods

This study utilizes administrative data from the Department of Statistics, Faculty of Science, for the first semester of the 2025 academic year. We developed the mathematical model using the Python programming language and the Gurobi optimization solver (Gurobi Optimization, 2025).

### Data Collection and Department Context

The scheduling problem encompasses all undergraduate and graduate programs within the department, including the Bachelor of Science in Statistics (STAT), the Bachelor of Science in Information Statistics and Data Science (SIDS), and the Master of Science in Applied Statistics. We derived the model's input parameters directly from the department's operational requirements through administrative interviews, resource verification, and faculty surveys.

Initial consultations with the faculty member responsible for timetabling established the baseline for the manual scheduling process and identified specific constraint requirements. Following this, we collaborated with administrative staff to compile physical and human resource specifications. The study incorporates seven specific classrooms, each defined by its maximum capacity and functional designation as a lecture room and/or a laboratory. These capacity limits serve as hard constraints for course assignment.

The model manages timetables for 13 full-time lecturers and a total enrollment of approximately 300 students. To prevent schedule overlaps for required courses, we segmented the student population by major and year level into nine distinct cohorts: STAT (Years 1-4), SIDS (Years 1-4), and Master's students. The operating hours run from Monday through Friday from 08:00 to 18:00. We modeled this duration as a set of discrete 30-minute timeslots, resulting in 18 periods per day. The interval from 12:00 to 13:00 is strictly reserved for lunch, prohibiting any course assignments during this hour.

Finally, we collected faculty teaching preferences via digital forms to determine availability. To maximize faculty satisfaction, we quantified these preferences into a numerical weight matrix for the objective function. A weight of 1.0 represents an instructor's ideal teaching time, 0.5 indicates a timeslot where the instructor is

available if necessary, and 0.0 signifies that the instructor is unavailable due to other administrative or personal commitments.

### Mathematical Formulation

The proposed ILP model is formulated to maximize a weighted sum of teaching preference while satisfying a set of hard constraints derived from departmental regulations. We define the notations of sets, parameters, and variables as in Table 1.

### Objective Function and Constraints

The objective function maximizes the total weighted preference score for all assigned courses. It is defined as

$$\max_{c \in C} \sum_{c \in C} \sum_{r \in R} \sum_{d \in D} \sum_{t \in T} \left( \frac{\sum_{p \in P} \text{Pref}_{pdt}}{2 \cdot \sum_{i \in I_c} \text{Hour}_{c,i}} \right) x_{crdt}$$

The term inside the parentheses represents the normalized preference weight. Since a course  $c$  spans multiple timeslots (specifically, twice its hourly duration) and may need multiple blocks in a week, summing the raw preference values would inflate the score for longer courses. Therefore, we introduce the scaling factor  $\frac{1}{2 \sum_{i \in I_c} \text{Hour}_{c,i}}$ . This normalization ensures that the contribution of any single assigned course to the objective value remains bounded between 0 and the minimum teaching preference of the assigned professor(s) for the corresponding assigned timeslots, regardless of the course's duration.

The mathematical architecture of the model is defined by a series of linear inequalities that translate administrative rules into hard constraints. Equations (1) - (3) establish the fundamental conflict-prevention layer for rooms, faculty, and cohorts. Equations (4) - (12) govern session integrity, room capacity, and the mandatory non-consecutive day gap. To ensure pedagogical continuity, Equations (13) - (21) utilize a flow-based formulation to track and enforce consecutive timeslots for single sessions. Finally, Equations (22) - (30) manage human resource limits, strictly regulating student attendance and faculty workload to prevent academic fatigue. Collectively, these 30 constraint sets process over 185,000 linear constraints to identify the global optimal schedule. A comprehensive mapping of these real-world rules to their corresponding mathematical constraints is provided in Table 2.

**Table 1. Summary of mathematical notation and definitions for the proposed formulation**

| Symbols                                                       | Descriptions                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sets and Indices</b>                                       |                                                                                                                                                                                                                                                                                                           |
| $C, I_c$                                                      | All courses offered (index $c$ ) and sessions for course $c$ (index $i$ ).                                                                                                                                                                                                                                |
| $R, R_c$                                                      | All classrooms (index $r$ ) and rooms eligible for course $c$ .                                                                                                                                                                                                                                           |
| $P, P_c, C_p$                                                 | Professors (index $p$ ), professors teaching course $c$ , and courses assigned to professor $p$ .                                                                                                                                                                                                         |
| $S, C_s$                                                      | All students (index $s$ ) and courses registered by student $s$ .                                                                                                                                                                                                                                         |
| $G, S_g, C_g$                                                 | Student cohorts (e.g., 1 <sup>st</sup> year STAT, 2 <sup>nd</sup> year SIDS) (index $g$ ), students in cohort $g$ , and courses for cohort $g$ .                                                                                                                                                          |
| $D, T$                                                        | Teaching days $D = \{1, 2, \dots, 5\}$ (index $d$ ) and timeslots $T = \{1, 2, \dots, 18\}$ (index $t$ ). Each slot represents a 30-minute teaching period scheduled between 08:00 and 18:00. The timeline excludes the one-hour lunch break from 12:00 to 13:00, which is between the 8th and 9th slots. |
| <b>Parameters</b>                                             |                                                                                                                                                                                                                                                                                                           |
| Hour <sub><math>c,i</math></sub>                              | Duration (in hours) required for session $i$ of course $c$ .                                                                                                                                                                                                                                              |
| Pref <sub><math>pdt</math></sub>                              | Preference weight (0, 0.5, 1) for professor $p$ on day $d$ at time $t$ .                                                                                                                                                                                                                                  |
| Cap <sub><math>r</math></sub> , Cap <sub><math>c</math></sub> | Capacity of room $r$ and student enrollment for course $c$ .                                                                                                                                                                                                                                              |
| A <sub><math>sdt</math></sub>                                 | 1 if student $s$ is available on day $d$ at time $t$ ; 0 otherwise.                                                                                                                                                                                                                                       |
| SlotLimit <sub>course</sub> <sup>daily</sup>                  | Maximum allowable assigned timeslots per day of all courses.                                                                                                                                                                                                                                              |
| SlotLimit <sub>student</sub> <sup>daily</sup>                 | Maximum allowable study timeslots per day of all students.                                                                                                                                                                                                                                                |
| SlotLimit <sub>professor</sub> <sup>daily</sup>               | Maximum allowable teaching timeslots per day of all professors.                                                                                                                                                                                                                                           |
| CourseLimit <sub>professor</sub> <sup>daily</sup>             | Maximum allowable teaching courses per day of all professors.                                                                                                                                                                                                                                             |
| NumCourse <sub><math>pd</math></sub> <sup>External</sup>      | Number of external courses professor $p$ teach on day $d$ .                                                                                                                                                                                                                                               |
| <b>Decision Variables</b>                                     |                                                                                                                                                                                                                                                                                                           |
| $x_{crdt}$                                                    | 1 if course $c$ is assigned to room $r$ on day $d$ at timeslot $t$ .                                                                                                                                                                                                                                      |
| $y_{pdt}$                                                     | 1 if professor $p$ is teaching on day $d$ at timeslot $t$ .                                                                                                                                                                                                                                               |
| $z_{scrdt}$                                                   | 1 if student $s$ attends course $c$ in room $r$ on day $d$ at timeslot $t$ .                                                                                                                                                                                                                              |
| $w_c, w_{cd}, w_{crd}$                                        | Indicators for course assignment: globally, on day $d$ , or in room $r$ on day $d$ .                                                                                                                                                                                                                      |
| $v_{cdt}, v_{cd}^{\max}$                                      | Consecutive periods count ending at $t$ ; Maximum consecutive chain on day $d$ .                                                                                                                                                                                                                          |
| $b_{cdt}$                                                     | 1 if the maximum consecutive periods for course $c$ on day $d$ end at timeslot $t$ .                                                                                                                                                                                                                      |

**Table 2. Mapping of real-world rules to mathematical constraints**

| Constraint Set      | Equation(s) | Real-World Meaning                                                                                                        |
|---------------------|-------------|---------------------------------------------------------------------------------------------------------------------------|
| Conflict Prevention | (1)-(3)     | Ensure that no room, professor, or student cohort is assigned to more than one course during any single timeslot          |
| Session Consistency | (4)-(5)     | Guarantee that a course session remains in the same room for its entire duration without switching locations mid-class    |
| Room Capacity       | (6)         | Ensure the assigned classroom's capacity is sufficient to accommodate the registered student enrollment                   |
| Lunch Break Spacing | (7)         | Prohibit any single session from bridging the morning and afternoon blocks                                                |
| Daily Integrity     | (8)-(9)     | Ensure that if a course is scheduled on a specific day, it occupies exactly the required number of hours for that session |
| Study Gap           | (10)        | Mandate at least a one-day gap between sessions of the same course                                                        |
| Course Completion   | (11)-(12)   | Guarantee that every course is fully scheduled for its total required weekly credit hours across the available days       |
| Block Continuity    | (13)-(21)   | Use flow-based logic to force multi-hour lectures to be scheduled in contiguous blocks                                    |
| Student Welfare     | (22)-(25)   | Prevent scheduling during known student unavailability and strictly caps daily study hours to prevent academic fatigue    |
| Faculty Workload    | (26)-(30)   | Respect instructor unavailability and cap both daily teaching hours and the number of distinct courses taught per day     |

The primary conflict-prevention constraint set is defined as follows. First, we enforce room exclusivity. No classroom may host more than one course simultaneously

$$\sum_{c \in C} x_{crdt} \leq 1 \quad \forall r \in R, d \in D, t \in T$$

(1)

Second, we enforce faculty availability. No professor may teach more than one course during the same timeslot

$$\sum_{c \in C_p} \sum_{r \in R} x_{crdt} \leq 1 \quad \forall p \in P, d \in D, t \in T.$$

(2)

Finally, we enforce strict cohort scheduling. All students in the same cohort must study at most one course simultaneously

$$\sum_{c \in C_g} \sum_{r \in R} x_{crt} \leq 1 \quad \forall g \in G, s \in S_g, d \in D, t \in T. \quad (3)$$

Collectively, inequalities (1), (2), and (3) serve as the model's primary conflict checks. By restricting the sum of binary decision variables on the left-hand side to at most 1, they ensure that for any specific timeslot  $t$  on day  $d$ , a room, a professor, or students in the same cohort are assigned to at most a single active course.

Many courses require multiple teaching sessions spread across the week (e.g., separate lectures and laboratories). While these distinct sessions may occupy different rooms on different days, it is critical that any single daily session remains in one location to avoid disrupting the flow of the class. To enforce this, we introduce an auxiliary binary variable  $w_{crt}$  to indicate if course  $c$  is assigned to room  $r$  on day  $d$ . This variable aggregates the granular timeslots to ensure that any daily assignment adheres to the specific time requirements of the course session  $i$ .

$$w_{crt} \leq \sum_{t \in T} x_{crt} \quad \forall c \in C, r \in R_c, d \in D. \quad (4)$$

$$\sum_{t \in T} x_{crt} = (2 \cdot \text{Hour}_{c,i}) \cdot w_{crt} \quad \forall c \in C, i \in I_c, r \in R_c, d \in D. \quad (5)$$

Equations (4) and (5) together link these indicators to the decision variables. A course is considered active on a specific day only if it has assigned timeslots, and the total duration on that day must match the course's required daily periods. Equation (4) mandates that a room is occupied for the day ( $w_{crt} = 1$ ) only if actual teaching time is scheduled within it ( $\sum_{t \in T} x_{crt} \geq 1$ ). Building on this, Equation (5) strictly enforces the duration. It dictates that if a course is assigned ( $w_{crt} = 1$ ), it must occupy exactly the required number of periods for session  $i$ . Together, these constraints guarantee that a course is fully scheduled for its required length in a single room, without switching locations mid-session. The next constraint set enforces a physical capacity constraint to guarantee that

room capacity is sufficient for course enrollment, expressed by

$$\text{Cap}_c \cdot w_{crt} \leq \text{Cap}_r \cdot w_{crt} \quad \forall c \in C, r \in R_c, d \in D. \quad (6)$$

This inequality dictates that if a course is assigned to a specific room ( $w_{crt} = 1$ ), the room's capacity must meet or exceed the course size. If the course is larger than the room, the inequality forces the variable  $w_{crt}$  to zero, thereby preventing the assignment.

To prevent fatigue and ensure optimal learning retention, we enforce spacing constraints. Courses are strictly prohibited from bridging the morning and afternoon blocks (spanning the lunch hour between 12:00 to 13:00), and they may not be scheduled on consecutive days (e.g., Monday and Tuesday), expressed by

$$x_{crt_8} + x_{crt_9} \leq 1 \quad \forall c \in C, r \in R_c, d \in D. \quad (7)$$

Equation (7) ensures that if course  $c$  is assigned on day  $d$ , it cannot span the lunch break nor occupy both timeslots 8 and 9.

We introduce auxiliary binary variables  $w_{cd}$  to indicate whether course  $c$  is assigned on day  $d$ , expressed by

$$w_{cd} \leq \sum_{r \in R} \sum_{t \in T} x_{crt} \quad \forall c \in C, d \in D, \quad (8)$$

$$\sum_{r \in R} \sum_{t \in T} x_{crt} = 2 \cdot \text{Hour}_{c,i} \cdot w_{cd} \quad \forall c \in C, d \in D. \quad (9)$$

Equation (8) constraints  $w_{cd}$  to be zero if course  $c$  is not assigned on day  $d$  ( $\sum_{r \in R} \sum_{t \in T} x_{crt} = 0$ ). While Equation (9) enforces that if course  $c$  is assigned to day  $d$  ( $w_{cd} = 1$ ), it must occupy exactly the required number of 30-minute intervals corresponding to its credit hours. Conversely, if  $w_{cd} = 0$ , no timeslots may be allocated.

To guarantee at least a one-day gap between sessions for the same course, we impose a constraint that prevents consecutive daily assignments. This arrangement provides students with adequate time for self-study and homework.

$$w_{cd} + w_{c(d+1)} \leq 1 \quad \forall c \in C, d \in \{1, 2, 3, 4\}. \quad (10)$$

Equation (10) enforces this by ensuring that the sum of assignment variables for any two adjacent days,  $d$  and  $d + 1$ , does not exceed one.

To ensure curriculum consistency, we define a binary decision variable  $w_c$  which indicates whether course  $c$  is included in the final schedule.

$$\sum_{r \in R} \sum_{d \in D} \sum_{t \in T} x_{crdt} = 2 \sum_{i \in I_c} \text{Hour}_{c,i} \cdot w_c \quad \forall c \in C. \quad (11)$$

$$\sum_{d \in D} w_{cd} = |I_c| \cdot w_c \quad \forall c \in C. \quad (12)$$

Equations (11) and (12) enforce an exact workload requirement that if course  $c$  is assigned ( $w_c = 1$ ), the total number of scheduled 30-minute intervals must match its weekly credit hours ( $2 \sum_{i \in I_c} \text{Hour}_{c,i}$ ). Conversely, if unassigned, no time can be allocated. Finally, Equation (12) ensures the course is distributed across the required number of distinct days ( $|I_c|$ ).

A critical requirement for pedagogical continuity is that all timeslots for a single session must be contiguous, a 3-hour lecture cannot be split into fragmented 1-hour blocks. To enforce this, we enforce consecutive period constraint. We implement a flow-based formulation using auxiliary variable  $v_{cdt}$  which counts the number of consecutive periods of course  $c$  assigned on day  $d$  ending at timeslot  $t$ , defined by

$$v_{cdt} \geq -t \cdot \sum_{r \in R} x_{crdt} \quad \forall c \in C, d \in D, t \in T, \quad (13)$$

$$v_{cdt} \leq t \cdot \sum_{r \in R} x_{crdt} \quad \forall c \in C, d \in D, t \in T, \quad (14)$$

$$v_{cdt_0} = 0 \quad \forall c \in C, d \in D, \quad (15)$$

$$v_{cdt} \geq 1 + v_{cd(t-1)} - t \cdot \left(1 - \sum_{r \in R} x_{crdt}\right) \quad \forall c \in C, d \in D, t \in T, \quad (16)$$

$$v_{cdt} \leq 1 + v_{cd(t-1)} + t \cdot \left(1 - \sum_{r \in R} x_{crdt}\right) \quad \forall c \in C, d \in D, t \in T. \quad (17)$$

Collectively, Equations (13) to (17) define  $v_{cdt}$ . If course  $c$  is not assigned on day  $d$  at timeslot  $t$  ( $\sum_{r \in R} x_{crdt} = 0$ ), Equations (13) and (14) force  $v_{cdt}$  to be zero. However, if course  $c$  is assigned on day  $d$  at timeslot  $t$  ( $\sum_{r \in R} x_{crdt} = 1$ ), the recursive relation in Equations (16) and (17) force  $v_{cdt}$  equals to  $1 + v_{cd(t-1)}$ , with a base case defined by Equation (15).

We then constrain the maximum value of this counter  $v_{cd}^{\max}$  to equal the total consecutive assigned periods of course  $c$  on day  $d$  or  $\max_{t \in T} v_{cdt}$ . If the assigned slots were fragmented, the counter would reset, and the maximum chain length would be less than the total duration, violating the equality. We define auxiliary binary variables  $b_{cdt}$  equal 1 if the maximum consecutive assigned periods of course  $c$  on day  $d$  end at timeslot  $t$ .

$$\sum_{t \in T} b_{cdt} = w_{cd} \quad \forall c \in C, d \in D. \quad (18)$$

$$v_{cd}^{\max} = \sum_{t \in T} \sum_{r \in R} x_{crdt} \quad \forall c \in C, d \in D. \quad (19)$$

$$v_{cd}^{\max} \geq v_{cdt} \quad \forall c \in C, d \in D, t \in T. \quad (20)$$

$$v_{cd}^{\max} \leq v_{cdt} + \text{SlotLimit}_{\text{course}}^{\text{daily}} \cdot (1 - b_{cdt}) \quad \forall c \in C, d \in D, t \in T. \quad (21)$$

Equations (18) to (21) together ensure that the nonnegative integer auxiliary variables  $v_{cd}^{\max}$  are the maximum of  $v_{cdt}$ , the number of consecutive periods of course  $c$  assigned on day  $d$  ending at timeslot  $t$ . Equation (18) enforces that if course  $c$  is assigned on day  $d$  ( $w_{cd} = 1$ ), there is exactly one timeslot  $t$  that the maximum consecutive periods end. Equation (19) defines the values of  $v_{cd}^{\max}$ . Finally, Equations (20) and (21) ensure the bound of  $v_{cd}^{\max}$ , and if the longest length of consecutive periods end at timeslot  $t$  ( $b_{cdt} = 1$ ), then  $v_{cd}^{\max} = v_{cdt}$ .

The final set of constraints ensures that the schedule remains within the physical and mental limits of the faculty and student body. To guarantee

full attendance, we link the course schedule variable  $x_{crdt}$  to the student attendance variable  $z_{srdt}$  where  $z_{srdt}$  are binary variables indicating whether student  $s$  scheduled to learn course  $c$  in room  $r$  on day  $d$  at timeslot  $t$ .

$$x_{crdt} \leq z_{srdt} \quad \forall s \in S, c \in C_s, r \in R_c, d \in D, t \in T. \quad (22)$$

$$x_{crdt} \leq A_{sdt} \quad \forall s \in S, c \in C_s, r \in R_c, d \in D, t \in T. \quad (23)$$

$$\sum_{c \in C} \sum_{r \in R} z_{srdt} \leq A_{sdt} \quad \forall s \in S, d \in D, t \in T. \quad (24)$$

$$\sum_{c \in C} \sum_{r \in R} \sum_{t \in T} z_{srdt} \leq \text{SlotLimit}_{\text{student}}^{\text{daily}} \quad \forall s \in S, d \in D. \quad (25)$$

Equation (22) dictates that course  $c$  can only be scheduled on day  $d$  at timeslot  $t$  ( $x_{crdt} = 1$ ) if the registered student  $s$  is able to attend ( $z_{srdt} = 1$ ). While, Equations (23) and (24) ensure that student  $s$  cannot attend on day  $d$  at timeslot  $t$  ( $x_{crdt} = \sum_{c \in C} \sum_{r \in R} z_{srdt} = 0$ ) if they are marked as unavailable in the data ( $A_{sdt} = 0$ ). Finally, the explicit tracking of student attendance via  $z_{srdt}$  allows us to regulate the daily academic load. Equation (25) enforces a maximum study duration, ensuring that the total number of timeslots student  $s$  is occupied on day  $d$  does not exceed a specified limit  $\text{SlotLimit}_{\text{student}}^{\text{daily}}$  (e.g., 6 hours or 12 slots). This prevents students from being overwhelmed by an excessive number of classes in a single day.

To manage faculty workload and availability, we introduce a set of constraints that track professor schedules, enforce availability, and limit daily teaching burdens. We limit the teaching load for every professor  $p$ . This includes a cap on the maximum teaching timeslots per day ( $\text{SlotLimit}_{\text{professor}}^{\text{daily}}$ ) and a limit on the number of distinct courses taught per day  $\text{CourseLimit}_{\text{professor}}^{\text{daily}}$ . We define an auxiliary binary variable  $y_{pdt}$  to indicate whether professor  $p$  teaches on day  $d$  at timeslot  $t$ .

$$\sum_{c \in C_p} \sum_{r \in R} x_{crdt} = y_{pdt} \quad \forall p \in P, d \in D, t \in T. \quad (26)$$

$$x_{crdt} \leq 2 \cdot \text{Pref}_{pdt} \quad \forall c \in C, p \in P_c, r \in R_c, d \in D, t \in T. \quad (27)$$

$$x_{crdt} = 0 \quad \forall c \in C, r \notin R_c, d \in D, t \in T. \quad (28)$$

$$\sum_{t \in T} y_{pdt} \leq \text{SlotLimit}_{\text{professor}}^{\text{daily}} \quad \forall p \in P, d \in D. \quad (29)$$

$$\text{NumCourse}_{pd}^{\text{External}} + \sum_{c \in C_p} w_{cd} \leq \text{CourseLimit}_{\text{professor}}^{\text{daily}} \quad \forall p \in P, d \in D \quad (30)$$

Equation (26) defines the binary variable  $y_{pdt}$ , which equals 1 if professor  $p$  is teaching any course in the department at time  $t$  on day  $d$ . Equation (27) strictly prohibits scheduling if the professor is marked as unavailable ( $\text{Pref}_{pdt} = 0$ ) with a weight factor of two to adjust when  $\text{Pref}_{pdt} = 0.5$ . We additionally set  $x_{crdt}$  to zero if  $r \notin R_c$  in Equation (28). Finally, we regulate faculty fatigue through workload limits. Equation (29) caps the total teaching duration per day to  $\text{SlotLimit}_{\text{professor}}^{\text{daily}}$  slots. Equation (30) limits the number of distinct courses a professor handles per day, taking into account both the variable departmental courses  $w_{cd}$  and any fixed external courses  $\text{NumCourse}_{pd}^{\text{External}}$  that are pre-assigned outside this optimization model.

## Results and Discussion

The optimization model was implemented in Python using Gurobi Optimizer 13.0.0 on an Intel Xeon 2.20GHz workstation. The dataset from the Department of Statistics in the academic semester 2025/1, comprised 13 professors, 7 classrooms (including 2 dedicated computer laboratories and 2 rooms reserved for graduate students), and 27 active courses. This framework integrates Python's extensive data processing ecosystem with Gurobi, which is recognized as a top-tier MILP solver (Mittelmann, 2024) and is available freely for academic research.

The resulting ILP formulation generated a large-scale problem instance. The solver processed a matrix consisting of 185,184 linear constraints and 605,565 variables, of which 602,865 were binary integer variables. Despite the combinatorial complexity inherent in NP-hard

scheduling problems, the model demonstrated high computational efficiency. A feasible solution was identified rapidly within 16 seconds. The solver successfully converged to the global optimal solution in 285.15 seconds. The final solution achieved a MIP gap of 0%, guaranteeing that the generated timetable is mathematically the best possible configuration based on the defined objective function.

The model successfully assigned all 27 offered courses to valid timeslots and rooms without any conflicts. The optimized objective value was 24.21. Given that there are 27 courses and the maximum preference weight for a perfect assignment is 1.0, this score indicates a high level of faculty satisfaction. On average, the schedule achieved a preference satisfaction rate of approximately 89.6%, meaning the vast majority of courses were assigned to the instructors' Preferred timeslots rather than merely Acceptable ones.

The generated timetable strictly adhered to all defined parameters that we initially set. No professor exceeded the maximum teaching daily limit of 3 courses, nor the maximum teaching limit of 12 timeslots (6 hours). Student cohorts were subjected to a maximum study load of 6.5 hours per day. The visual inspection of the output tables confirms a break between timeslots 8 and 9 as shown in an example in Figure 1.

The results show major improvements compared to manual scheduling. Previously, staff needed several days to adjust the schedule and fix conflicts. In contrast, this new model created a perfect, conflict-free schedule in under five minutes. The model also automatically followed complex rules that people often miss when planning by hand. For example, it kept long lectures together in one continuous block and ensured courses were not scheduled on days right next to each other.



**Figure 1. Comparison of example scheduling outputs. The top panel shows the optimal result from the proposed full model. The middle panel shows the result without non-consecutive day constraints, and the bottom panel illustrates the fragmentation caused by removing consecutive period constraints**

To evaluate the performance and necessity of the proposed constraint formulations, we conducted a systematic comparison between the full model and two baseline variants. The quantitative results are summarized in Table 3, and the qualitative impact on schedule structure is visualized in Figure 1. The top panel of the Figure 1 demonstrates the full model's ability to maintain coherent, continuous blocks with adequate daily spacing. In contrast, the baseline without consecutive-period tracking (Equations (13)-(21) removed), shown in the bottom-right panel, solved faster (10.85s) and achieved a higher preference score (93.6%) but at the severe cost of fragmenting intensive courses into disjointed, non-contiguous blocks that disrupt instructional flow. Similarly, the baseline without the non-consecutive-day constraint (Equation (10) removed), depicted in the bottom-left panel of Figure 1, produced schedules where courses are clustered on adjacent days, without necessary study gaps. Quantitatively, this day-gap baseline achieved a score of 91.13% but required a significantly longer solve time (496.82s), illustrating that tight constraints can actually accelerate optimization by effectively pruning the feasible search space. These results demonstrate that while our full constraint set increases computational complexity relative to the simplest baseline, it is essential for enforcing critical educational quality requirements that cannot be satisfied through post-hoc manual adjustment.

**Table 3. Computational results comparing the proposed full model against baselines without non-consecutive day constraints and without consecutive period tracking**

| Model Variant                | Time (s) | MIP Gap | Preference Score |
|------------------------------|----------|---------|------------------|
| Full Model                   | 285.15   | 0%      | 24.21 (89.6%)    |
| Without Non-Consecutive Days | 496.82   | 0%      | 24.60 (91.13%)   |
| Without Consecutive Periods  | 10.85    | 0%      | 25.27 (93.6%)    |

This method is flexible enough to handle many real-life situations beyond just solving conflicts. By adding simple constraints, the model can accommodate specific needs like joint classes or fixed time slots. It can also manage facility requirements, such as assigning graduate courses to specific seminar rooms. Finally, the system helps manage faculty workloads. It respects when instructors are unavailable and balances their daily teaching hours to prevent them from getting burnt out.

However, this flexibility is ultimately bounded by physical resource limits. While the proposed model efficiently solves the current departmental dataset to global optimality, it is crucial to understand the implications of more difficult scheduling scenarios. In timetabling, difficulty is often a function of resource saturation rather than computational complexity alone. If the model were applied to a scenario with significantly more restrictive constraints, such as reduced room availability, tighter faculty workload caps, or highly specific time preferences, the solution space would contract. This would result in either a lower global preference score or, in extreme cases of resource deficit, a status of infeasibility. Thus, the primary challenge in scaling this approach is not necessarily solver runtime, but ensuring that the physical and administrative constraints allow for a non-empty feasible region.

## Conclusions

This study presented an ILP approach to solve the UCTTP for the Department of Statistics at Khon Kaen University. By modeling the scheduling requirements as a mathematical optimization problem, we successfully automated a process that is traditionally labor-intensive and error-prone. The proposed model distinguishes itself by incorporating pedagogical constraints often absent in standard formulations, specifically the consecutive period constraint to prevent fragmented lectures and the non-consecutive day constraint to optimize student study intervals. Tested against the Semester 1/2025 curriculum involving 27 courses and 13 faculty members, the model solved to global optimality in 285.15 seconds, generating a complete, conflict-free timetable that maximized faculty teaching preferences while strictly adhering to physical resource limits and human workload capacities.

Despite these promising results, several limitations must be acknowledged. First, the ILP approach is computationally tractable primarily for departmental-scale problems, with approximately 30-50 courses; university-wide scheduling with hundreds of courses would require decomposition strategies or hybrid methods that may sacrifice global optimality guarantees. Second, the model relies on static preferences collected at the start of the semester, meaning dynamic re-optimization to address mid-semester faculty availability changes is not currently supported. Third, the current formulation maximizes a single weighted

preference score, whereas real institutional contexts often require multi-criteria optimization to explicitly balance conflicting stakeholder goals, such as faculty convenience versus room utilization efficiency. Finally, the model relies on deterministic assumptions regarding input data and does not currently model stochastic variations in actual student enrollments.

Future research will address these challenges through four primary avenues. To enhance scalability, we aim to implement decomposition techniques, such as Benders decomposition or column generation, to enable multi-department scheduling while preserving solution quality guarantees. Crucially, we aim to improve fairness by incorporating stable matching to prevent instructors with restrictive preference from disproportionately securing preferred slots at the expense of flexible colleagues. We also plan to develop a web-based decision support interface allowing administrative staff to adjust parameters in real-time, visualize constraint violations, and observe scheduling trade-offs immediately. Furthermore, future iterations will explore multi-objective formulations, such as weighted goal programming, to generate Pareto-optimal schedules, alongside robust optimization techniques to handle enrollment uncertainty.

Ultimately, these results demonstrate that exact methods like ILP are not only theoretically sound but practically viable for departmental-level scheduling. The transition to this automated system allows administrative staff to shift their focus from solving logical conflicts to strategic academic planning.

## Funding

The authors received no specific financial support for the research, authorship, and/or publication of this article.

## Author Contributions

Conceptualization: P.W., S.P.; Methodology: P.W., S.P.; Investigation: P.W., N.S., Y.S.; Data Curation: P.W., N.S., Y.S.; Writing – Original Draft: P.W.; Writing – Review & Editing: P.W., S.P.; Supervision: P.W. All authors have read and agreed to the published version of the manuscript.

## Ethics Statement

Ethical approval was not required for this study because it did not involve human participants, animals, identifiable personal data, or materials requiring institutional ethical review.

## Declaration of Generative AI Use

During the preparation of this manuscript, the authors used a generative AI solely for language refinement and typo checking of text originally written entirely by the authors. No AI tool was used to generate content, analysis, or code. All AI-assisted output was critically reviewed, verified, and revised by the authors, who accept full responsibility for the content of this manuscript.

## Conflict of Interest

The authors declare that they have no conflicts of interest related to this work.

## Data Availability Statement

The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

## Acknowledgement

We would like to thank the staff of the Department of Statistics, Khon Kaen University, for their assistance. Special thanks are due to Assistant Professor Dr. Tidadeaw Mayureesawan for her invaluable insights and work on the manual scheduling process.

## References

- Alvarez-Valdes, R., Crespo, E., & Tamarit, J. M. (2002). Design and implementation of a course scheduling system using tabu search. *European Journal of Operational Research*, 137(3), 512-523. [https://doi.org/10.1016/S0377-2217\(01\)00091-1](https://doi.org/10.1016/S0377-2217(01)00091-1)
- Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), 266-280. [https://doi.org/10.1016/S0377-2217\(02\)00069-3](https://doi.org/10.1016/S0377-2217(02)00069-3)

- Burke, E., Elliman, D., & Weare, R. (1995). A hybrid genetic algorithm for highly constrained timetabling problems. In *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95)* (pp. 605-610). Morgan Kaufmann.
- Cooper, T. B., & Kingston, J. H. (1995). The complexity of timetable construction problems. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 281-295). Springer. [https://doi.org/10.1007/3-540-61794-9\\_66](https://doi.org/10.1007/3-540-61794-9_66)
- Elmohamed, M. S., Coddington, P., & Fox, G. (1997). A comparison of annealing techniques for academic course scheduling. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 92-112). Springer. <https://doi.org/10.1007/BFb0055883>
- Gurobi Optimization, LLC. (2025). *Gurobi optimizer reference manual* (Version 13.0). Gurobi Optimization.
- Harrabi, O., Siala, J. C., & Mrad, M. (2024). An optimisation-based system for the university course timetabling: A novel integer linear programming model. *International Journal of Industrial and Systems Engineering*, 46(2), 195-214. <https://doi.org/10.1504/IJISE.2024.136412>
- Land, A. H., & Doig, A. G. (2009). An automatic method for solving discrete programming problems. In *50 years of integer programming 1958-2008: From the early years to the state-of-the-art* (pp. 105-132). Springer. [https://doi.org/10.1007/978-3-540-68279-0\\_5](https://doi.org/10.1007/978-3-540-68279-0_5)
- Mallari, C. B., San Juan, J. L., & Li, R. (2023). The university coursework timetabling problem: An optimization approach to synchronizing course calendars. *Computers & Industrial Engineering*, 184, 109561. <https://doi.org/10.1016/j.cie.2023.109561>
- Mittelmann, H. D. (2024). *Decision tree for optimization software*. Arizona State University.
- Mokhtari, M., Vaziri Sarashk, M., Asadpour, M., Saeidi, N., & Boyer, O. (2021). Developing a model for the university course timetabling problem: A case study. *Complexity*, 2021, 9940866. <https://doi.org/10.1155/2021/9940866>
- Müller, T., Rudová, H., & Müllerová, Z. (2025). Real-world university course timetabling at the International Timetabling Competition 2019. *Journal of Scheduling*, 28(2), 247-267. <https://doi.org/10.1007/s10951-023-00801-w>
- Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L. M., Knowles, J., Manfrin, M., Mastrolilli, M., Paechter, B., Paquete, L., & Stützle, T. (2003). A comparison of the performance of different metaheuristics on the timetabling problem. In E. Burke & P. De Causmaecker (Eds.), *Practice and theory of automated timetabling IV: PATAT 2002* (Lecture Notes in Computer Science, Vol. 2740, pp. 329-351). Springer. [https://doi.org/10.1007/978-3-540-45157-0\\_22](https://doi.org/10.1007/978-3-540-45157-0_22)
- Talmor, I. (2024). Optimizing academic timetables using integer linear programming: A case study. *Educational Sciences and Management*, 2(3), 176-187. <https://doi.org/10.56578/esm020305>