

The lecture-hall quadratic assignment problem

Tal Raviv^{a,*}

^a*School of Industrial & Intelligent Systems Engineering, Tel Aviv University, Tel Aviv, Israel*

Preprint, not peer reviewed yet. April 23, 2026

Abstract

We study the post-timetabling *lecture-hall assignment problem*, in which a fixed weekly timetable and realized student enrollments are given, and the decision is to assign each lecture to a compatible hall so as to minimize the walking burden imposed on students moving between consecutive lectures and a penalty for feasible but non-ideal compatibility of halls-lecture. The problem combines a quadratic assignment objective, driven by the product of shared enrollments and hall-to-hall distances, with interval-scheduling feasibility constraints and hall-capacity restrictions. We prove that the problem is NP-hard and that every approximation lower bound for the classical quadratic assignment problem carries over, already in a single-day, two-block special case. We develop three exact formulations: a bilinear mixed-integer quadratic program, a compact mixed-integer linear program with one auxiliary distance variable per successor pair, and a constraint-programming model. All of these formulations admitting a natural daily decomposition. To tighten the relaxations we introduce biclique distance cuts that dominate the standard pairwise linking inequalities, together with Chvátal-Gomory capacity-dominance cuts and a safe compatibility-preprocessing procedure. The framework also accommodates common timetabling side constraints such as hard and soft *Same-Room* and *SameAttendees* requirements, for which biclique-strengthened linearizations are derived. A comprehensive numerical experiment covers daily instances derived from five large ITC 2019 competition benchmarks and from a 2023 institutional dataset from Lancaster University. The compact linearized program emerges as the most robust formulation, and biclique strengthening is the dominant modeling device: it closes virtually all of the root-relaxation gap and lifts the number of proven-optimal solves substantially, while cutting median solver runtimes by roughly an order of magnitude. The results demonstrate that realistically sized lecture-hall assignment instances can be solved to optimality or near-optimality within minutes, and frequently even within seconds, by off-the-shelf mixed-integer programming solver once the biclique structure of the quadratic walking term is exploited.

Keywords: lecture-hall assignment, quadratic assignment problem, university course timetabling, interval scheduling, mixed-integer programming, constraint programming, biclique inequalities

1. Introduction and background

Universities and colleges often determine the course timetable well before they finalize the assignment of lectures to specific halls. Throughout this paper, we use the term *hall* (or *lecture-hall*) to refer generically to any space used for academic activity, whether it is a traditional lecture theater, a seminar room, or a laboratory. Similarly, we use the term *lecture* to refer to any scheduled academic activity, encompassing regular lectures, tutorials, or practical sessions. Once the timetable is fixed

*Corresponding author

Email address: talraviv@tauex.tau.ac.il (Tal Raviv)

and student registration is completed, the institution knows both the enrollment in each lecture and the enrollment overlap between consecutive lectures. At that stage, the hall-assignment decision can have a substantial effect on student mobility across campus. Assigning consecutive lectures attended by many of the same students to distant halls may generate unnecessary walking, even when every lecture is assigned to a feasible hall. In practice, long walks between halls may also cause students to arrive late at their next lecture, thereby disrupting the beginning of a lecture. It may also be particularly challenging to students with mobility disabilities.

This observation motivates a lecture-hall assignment problem that is separate from the timetabling itself. In the lecture-hall assignment problem, the timetable, including the scheduled time interval of each lecture, is given. The remaining task is to assign lectures to halls in a way that respects hall compatibility in terms of capacity and required facilities, while also reducing the walking burden imposed on students as they move between consecutive lectures. Other, more local, considerations may also affect the desirability of assignment of each lecture to a compatible hall, e.g., large halls are feasible but less desirable for lectures with low enrollment. In this paper, we model this problem and present exact solution methods for it.

The proposed model is related to three main streams of literature: the quadratic assignment problem (QAP), lecture-hall assignment and interval scheduling, and university course timetabling. The closest conceptual link to the QAP is the presence of pairwise interaction costs: assigning two lectures to two halls creates a cost that depends jointly on the distance between the halls and on the number of students who attend both lectures. This is analogous to the flow–distance structure that defines the classical QAP introduced by Koopmans and Beckmann (1957) and surveyed extensively by Burkard et al. (1998) and Loiola et al. (2007). The QAP literature has developed a rich toolkit of solution techniques whose ideas we draw on in this paper. Classical lower bounds begin with the Gilmore–Lawler bound (Gilmore, 1962; Lawler, 1963), which is obtained from a linear assignment relaxation of the quadratic cost structure. Linearization-based formulations were proposed by Kaufman and Broeckx (1978), who introduced a compact Benders-style reformulation using one auxiliary continuous variable per pair of facilities, and by Frieze and Yadegar (1983), whose linearization exploits the permutation structure. Adams and Johnson (1994) later developed a stronger mixed-integer linear reformulation whose linear-programming relaxation dominates earlier bounds, at the cost of a larger variable set. More recent advances include dual-based bounds derived from a Hungarian-method reformulation of the linearized model (Hahn and Grant, 1998), semidefinite relaxations (Zhao et al., 1998), lift-and-project branch-and-cut algorithms that solve increasingly tight binary relaxations on the fly (Burer and Vandenbussche, 2006), convex quadratic reformulations and improved branch-and-bound methods (Anstreicher, 2003), and large-scale computational studies such as Anstreicher et al. (2002), who solved previously intractable QAPLIB instances on computational grids. The same combinatorial backbone reappears in closely related facility-layout problems, for which Fischer et al. (2019) develop exact branch-and-cut algorithms based on layered mixed-integer program (MIP) and semidefinite relaxations and report state-of-the-art bounds on standard benchmarks. On the heuristic side, robust tabu search (Taillard, 1991) and related metaheuristics remain the dominant approach for obtaining high-quality solutions to large QAP instances. The compact linearization and the valid inequalities that we develop in Section 3 are inspired by this branch-and-cut line of work, in particular by the lift-and-project ideas of Burer and Vandenbussche (2006) and the layered linearization of Fischer et al. (2019), but are adapted to the combined temporal and compatibility structure of the lecture-hall setting and exploit the sparsity of the successor relation that is absent from the dense classical QAP. At the same time, the present problem differs from the standard QAP because feasibility is driven by temporal overlap, hall capacities, and the subset of lecture pairs that are consecutive on the same day.

A second relevant stream concerns lecture-hall assignment and interval-based hall allocation. In this family of problems, activities with fixed or partially fixed time intervals must be assigned to halls while respecting incompatibilities caused by overlap in time. Carter and Tovey (1992) study the lecture-hall assignment problem from a complexity perspective and explicitly connect it to interval scheduling. The broader interval-scheduling literature, surveyed by Kolen et al. (2007), studies machine or resource assignment of activities with fixed time windows and provides a natural framework for the temporal-feasibility layer of our problem. In particular, the clique structure of interval graphs, which we exploit in Section 3 to tighten the overlap constraints, is a standard building block in that literature. The framework of Carter and Tovey (1992) and Kolen et al. (2007) is closely related to the feasibility component of the present model, where lectures cannot share a hall if their time intervals overlap. However, the objective considered here is richer because it is not limited to feasibility or hall utilization; instead, it introduces interaction costs across consecutive lectures through student flows and hall-to-hall distances.

A third stream is the university timetabling literature, which studies the joint allocation of courses to times and halls under institutional constraints. This is a problem which the literature refers to as the *University Course Timetabling Problem*. General surveys by Schaefer (1999) and Burke and Petrovic (2002) describe a broad range of formulations, including problems in which hall capacities, conflicts, and soft preferences play a central role, while the more recent review of Abdipoor et al. (2023) summarizes modern meta-heuristic and hybrid approaches for university course timetabling. The field has also been shaped by three international timetabling competitions. The first competition (ITC 2002) established graph-coloring-style formulations and the first widely used benchmark set; Lewis and Paechter (2005) illustrate the graph-grouping heuristics that dominated the early competition and provide lower bounds based on the chromatic structure of the conflict graph. The second competition (McCollum et al., 2010) broadened the problem to three tracks, including curriculum-based course timetabling, for which Bonutti et al. (2012) later introduced the now-standard benchmark suite, data formats, and validation protocol. The ITC 2019 competition (Müller et al., 2025) further generalized the model by adding rich distribution constraints and student sectioning, and produced the dataset used in our real-world campaign. On the metaheuristic side, Abdullah and Turabieh (2012) develop a tabu-based memetic approach with multiple neighborhood structures that remains a reference point for high-quality feasible solutions on the curriculum-based competition instances. Within this literature, lecture-hall assignment is often treated either as a sub-problem of course timetabling or as a second-stage problem after times have been fixed. For example, Torres-Ovalle et al. (2014) propose a two-phase approach in which course scheduling is followed by lecture-hall assignment, and Rappos et al. (2022) develop a mixed-integer programming model that jointly considers student allocation, hall assignment, and time-period assignment. A recent extension by Davison et al. (2025) brings hybrid teaching into the university course timetabling problem by treating each lecture’s delivery mode: fully online, fully in-person, or hybrid with a split of students between the two modes, as a first-class modeling element alongside the usual hall and time decisions. They formulate a multi-objective binary program that combines standard timetabling features identified from a broad literature review with new variables and constraints describing in-person versus online delivery and student mode preferences, and they solve it by a lexicographic method on benchmark data, illustrating trade-offs among demand satisfaction, hall utilization, and hybrid-delivery quality for use in strategic decision making. Their work expands the set of constraints and objectives that a timetabling model is expected to handle but, like most models in this tradition, keeps hall assignment embedded inside the larger multi-criterion formulation rather than isolating it as a separate problem with exploitable structure. This decomposition is natural in many university settings and is also compatible with the present formulation, which assumes that the scheduled time interval of each lecture is already known.

Taken together, these three lines of literature suggest that the lecture-hall assignment problem studied here combines structural ingredients that are usually treated separately. From lecture-hall assignment and interval scheduling it inherits the temporal packing constraints on halls; from timetabling it inherits the academic context, capacity restrictions, and interpretation of lectures as scheduled activities; and from the QAP it inherits a pairwise objective that couples assignment decisions across lectures. A particularly close recent connection is the hall-allocation study of Antunes-Batista et al. (2026), who isolate the hall-assignment subproblem of the university course timetabling problem under an already-fixed weekly schedule and formulate it as a six-criterion problem with two hard constraints (hall capacity and no scheduling overlap) and four soft objectives (student relocation between buildings, mismatch between requested and assigned hall types, wasted capacity, and hall changes between consecutive lectures of the same student group). They develop customized multiobjective evolutionary algorithms augmented with problem-specific local search and constructive heuristics implemented in the jMetal framework, and benchmark them against an ILP model solved by Gurobi. The computational study uses a full-semester dataset from the University Institute of Lisbon with approximately 10,000 students, more than 26,000 scheduled lecture sessions, and over 120 lecture-halls of more than 20 distinct types, and emphasizes scalability and Pareto-front quality as the central performance measures.

The present framework can likewise accommodate mandatory hall features through the compatibility sets and additional hall-quality criteria through linear assignment penalties, so on the modeling surface it overlaps with four of the six criteria of Antunes-Batista et al. (2026). Its distinguishing features are structural and methodological. Structurally, student movement enters as a QAP-style quadratic term induced by shared enrollments and inter-hall walking distances rather than as a count of building changes or group-level hall changes; capacity and overlap are retained as hard feasibility constraints; and the remaining hall-quality criteria are combined into a single scalar objective through linear assignment penalties, which is possible because the weighting between them is under the analyst’s control. Methodologically, this combined structure motivates exact mathematical-programming and constraint-programming approaches with problem-specific valid inequalities and preprocessing, rather than the multiobjective evolutionary framework favored in Antunes-Batista et al. (2026). The two approaches are therefore complementary: theirs targets very large operational instances where a Pareto front among competing criteria is the primary deliverable, while ours targets a clean QAP-flavored subproblem where exploiting pairwise structure and interval-graph cliques often yields provably optimal or near-optimal solutions.

Research gap and contribution.. While the QAP literature provides powerful exact algorithms (Kaufman and Broeckx, 1978; Frieze and Yadegar, 1983; Adams and Johnson, 1994; Hahn and Grant, 1998; Zhao et al., 1998; Anstreicher et al., 2002; Anstreicher, 2003; Burer and Vandembussche, 2006; Fischer et al., 2019), it assumes unrestricted assignments without temporal overlap or capacity constraints. Conversely, the timetabling and interval-scheduling literatures address these operational realities but typically rely on linear objectives or multiobjective heuristics rather than QAP-style quadratic pairwise interactions (Carter and Tovey, 1992; Kolen et al., 2007; Lewis and Paechter, 2005; McCollum et al., 2010; Bonutti et al., 2012; Abdullah and Turabieh, 2012; Rappos et al., 2022; Davison et al., 2025; Antunes-Batista et al., 2026).

We bridge this gap by isolating the post-timetabling lecture-hall assignment problem, a setting common in institutions where course enrollments are only revealed after the schedule is set. In our framework, temporal overlap and spatial capacity act as hard constraints (with optional linear penalties for soft preferences), while the primary objective is a quadratic mapping of walking effort driven by shared student enrollments. By formally separating this subproblem, we exploit its distinct structure. We analyze its computational complexity and construct a novel, compact two-

index distance-variable formulation. To strengthen this model, we introduce an easy-to-separate valid biclique inequality that shares the strength of classical lift-and-project cuts but operates in a lower dimension by leveraging the sparsity of lecture-hall compatibilities. Through comprehensive numerical studies, we demonstrate that these problem-specific mathematical-programming and constraint-programming enhancements substantially improve computational performance over baseline nonlinear formulations, solving large, realistic instances to optimality or near-optimality.

2. Problem statement and complexity

We now formalize the lecture-hall assignment problem and state its complexity properties.

We consider this problem through the following sets:

Halls (H) Each hall $h \in H$ is defined by its capacity u_h , its location, and its set of available features F_h (such as “laboratory bench”, “computer workstation”, “projector”, or “wheelchair-accessible entrance”).

Lectures (L) Each lecture $l \in L$ is defined by its scheduled time interval, its number of registered students s_l , and its set of mandatory hall features F_l^{req} (for example, a chemistry practical must be assigned to a lab hall, and a programming session must be assigned to a hall with computer workstations).

Immediate-successor relation (A) A relation $A \subseteq L \times L$, where $(l_1, l_2) \in A$ if lecture l_2 is scheduled immediately after lecture l_1 for at least one group of students.

A hall h is *compatible* with a lecture l if it has sufficient capacity and provides every mandatory feature, that is,

$$H(l) = \{h \in H : u_h \geq s_l, F_l^{\text{req}} \subseteq F_h\}.$$

The compatibility set $H(l)$ is computed once from the input data and is treated as primitive in the rest of the paper. Mandatory features are deliberately handled as hard restrictions through $H(l)$ rather than as soft penalties, because in practice a chemistry practical cannot meaningfully take place in a hall without lab benches no matter what the student-flow gain would be.

The walking distance between each pair of halls $h_1, h_2 \in H$ is given by the distance matrix d_{h_1, h_2} . In addition, for each pair of lectures $(l_1, l_2) \in A$, we are given the number of students enrolled in both lectures, denoted by c_{l_1, l_2} . This number may be a simple head count or a score that gives higher weight to students with disabilities, if the institution has this information at its disposal. Finally, for each compatible lecture and hall pair (l, h) with $h \in H(l)$, we are given a nonnegative *assignment penalty* $p_{lh} \geq 0$ that captures local hall-quality considerations beyond the hard compatibility filter. In practice, p_{lh} is constructed by aggregating several soft per-assignment effects: wasted seating capacity when a small lecture is placed in a much larger hall, absence of desirable but non-mandatory features (for example, an oversized whiteboard desired by the instructor or a preferred building wing), and institutional preferences inherited from a higher-level timetabling stage. Any such weighting is encoded by adjusting the input p_{lh} before solving and does not change the formulations developed below; the relative importance of each component is therefore under the analyst’s control. Together, the three ingredients d_{h_1, h_2} , c_{l_1, l_2} , and p_{lh} capture the cross-lecture coupling driven by student movement and the per-assignment local quality of each hall choice.

We make the following standing assumption on the registration data.

Single-lecture-at-a-time assumption No student is registered for two lectures whose scheduled intervals overlap in time.

Under this assumption, every student contributes to the shared enrollment c_{l_1, l_2} of a successor pair $(l_1, l_2) \in A$ at most once, and the walking-distance objective defined in the next section is well-posed on every feasible assignment. Any input that violates the assumption must be cleaned beforehand by dropping the conflicting registration from one of the two overlapping lectures; Subsection 4.1 describes one such cleaning step in our Lancaster benchmark.

The goal is to assign each lecture to a compatible hall such that no two lectures assigned to the same hall overlap in time. The objective is to minimize the sum of two terms: (i) the total walking distance traveled by students between consecutive lectures, and (ii) the total assignment penalty incurred by the chosen lecture-hall pairs. The relative weight between the two terms is implicit in the units of c_{l_1, l_2} and p_{lh} , and is therefore controlled by the analyst when these inputs are constructed.

A natural decision version of the quadratic lecture-hall assignment problem asks whether, for a given bound B , there exists a feasible hall assignment whose total walking cost is at most B . This decision problem belongs to NP, since a proposed assignment can be verified in polynomial time by checking compatibility, non-overlap, and the resulting objective value. The next proposition states the exact and approximation hardness of the problem.

Proposition 1. *The decision version of the lecture-hall assignment problem is NP-complete, and the corresponding optimization problem is NP-hard. Moreover, for every $\rho \geq 1$, if there exists a polynomial-time ρ -approximation algorithm for the lecture-hall assignment problem, then there exists a polynomial-time ρ -approximation algorithm for the classical quadratic assignment problem (QAP). Consequently, every approximation lower bound for QAP carries over to the lecture-hall assignment problem. All of these statements remain true even for a restricted single-day case with two consecutive time blocks, identical hall capacities, and full lecture-to-hall compatibility.*

Proof. We use a reduction from the classical QAP, which is NP-hard and inapproximable within a constant factor, see Sahni and Gonzalez (1976); Burkard et al. (1998); Loiola et al. (2007). Consider a QAP instance with facilities $\mathcal{N} = \{1, \dots, n\}$, locations $\mathcal{R} = \{1, \dots, n\}$, nonnegative flow matrix (f_{ij}) , and nonnegative distance matrix (d_{ab}) . First normalize the distances by setting

$$\hat{d}_{aa} = 0 \quad \forall a \in \mathcal{R}, \quad \hat{d}_{ab} = d_{ab} + 1 \quad \forall a \neq b.$$

For every permutation π of the locations,

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \hat{d}_{\pi(i), \pi(j)} = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\pi(i), \pi(j)} + \sum_{i \neq j} f_{ij},$$

because $\pi(i) \neq \pi(j)$ whenever $i \neq j$. Thus the normalization adds the same constant to every feasible QAP solution, so it preserves both exact comparisons and approximation ratios.

Now construct a lecture-hall instance as follows. Create n halls h_1, \dots, h_n , all with identical capacity $u_{h_a} = U$ for some U larger than every lecture size introduced below, and declare the hall feature set $F_{h_a} = \emptyset$ for every a . For each lecture l introduced in the remainder of the construction, set the required-feature set $F_l^{\text{req}} = \emptyset$ and choose a registered-students count $s_l \leq U$, so that by the definition of compatibility in Section 2 every hall is compatible with every lecture. Set the assignment penalty $p_{lh} = 0$ for every lecture-hall pair so that the assignment-penalty term in the objective contributes nothing. Create two sets of lectures,

$$W = \{w_1, \dots, w_n\} \quad \text{and} \quad V = \{v_1, \dots, v_n\},$$

where all lectures in W occupy one common time block and all lectures in V occupy the immediately following common time block. Every lecture in W therefore overlaps every other lecture in W , and

similarly for V . Since there are exactly n halls, every feasible lecture-hall assignment induces two permutations of the halls: one permutation π for the lectures in W and one permutation σ for the lectures in V . For every $i, j \in \{1, \dots, n\}$, include the pair (w_i, v_j) in the successor set and set

$$d_{h_a, h_b} = \hat{d}_{ab} \quad \forall a, b \in \{1, \dots, n\}.$$

Let M be a positive penalty parameter, to be specified below, and define

$$c_{w_i, v_j} = \begin{cases} f_{ij} + M, & \text{if } i = j, \\ f_{ij}, & \text{if } i \neq j. \end{cases}$$

Under a feasible lecture-hall assignment corresponding to permutations (π, σ) , the objective value is

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \hat{d}_{\pi(i), \sigma(j)} + M \sum_{i=1}^n \hat{d}_{\pi(i), \sigma(i)}.$$

If $\pi \neq \sigma$, then there exists some i such that $\pi(i) \neq \sigma(i)$, and therefore $\hat{d}_{\pi(i), \sigma(i)} \geq 1$. Hence every such assignment has value at least M . On the other hand, if $\pi = \sigma$, then the penalty term vanishes and the lecture-hall objective becomes

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \hat{d}_{\pi(i), \pi(j)},$$

which is exactly the normalized QAP objective.

For the exact reduction, start from the QAP decision problem with threshold K and set

$$M = \hat{K} + 1, \quad \hat{K} := K + \sum_{i \neq j} f_{ij}.$$

Then any lecture-hall solution of value at most \hat{K} must satisfy $\pi = \sigma$, and once $\pi = \sigma$ its value is exactly the normalized QAP value of the same permutation. Therefore the original QAP instance is a yes-instance if and only if the constructed lecture-hall instance admits a feasible assignment of cost at most \hat{K} . The quantity \hat{K} is a sum of at most $n^2 + 1$ nonnegative integers taken directly from the QAP input, so its binary bit-length is polynomial in the bit-length of that input; consequently $M = \hat{K} + 1$ and the resulting walking weights $c_{w_i, v_j} \in \{f_{ij}, f_{ij} + M\}$ all fit in polynomially many bits, and the full lecture-hall instance is produced in polynomial time from the QAP input. Hence the construction is a Karp (polynomial-time many-one) reduction, not merely a Cook reduction, and the decision version of the lecture-hall assignment problem is NP-hard. Since it belongs to NP, it is NP-complete. Consequently, the optimization problem is NP-hard.

For the approximation statement, fix $\rho \geq 1$ and let

$$F_{\text{tot}} = \sum_{i=1}^n \sum_{j=1}^n f_{ij}, \quad D_{\text{max}} = \max_{a, b \in \mathcal{R}} \hat{d}_{ab}, \quad U_{\text{max}} = F_{\text{tot}} D_{\text{max}}.$$

Every feasible normalized QAP solution has value at most U_{max} , and therefore the optimal value OPT_{QAP} satisfies $\text{OPT}_{\text{QAP}} \leq U_{\text{max}}$. Now set

$$M = \rho U_{\text{max}} + 1.$$

For any fixed rational $\rho \geq 1$ (equivalently, for ρ given as part of the input in binary), $U_{\text{max}} = F_{\text{tot}} D_{\text{max}}$ is a product of two nonnegative integers whose binary bit-lengths are polynomial in the QAP input,

so M itself has bit-length polynomial in the input and the construction of the lecture-hall instance runs in polynomial time. The approximation-preserving reduction is therefore also Karp, not merely Cook. Then every lecture-hall solution with $\pi \neq \sigma$ has value at least

$$M = \rho U_{\max} + 1 > \rho U_{\max} \geq \rho \text{OPT}_{\text{QAP}}.$$

But the optimal lecture-hall value is exactly OPT_{QAP} , attained by solutions with $\pi = \sigma$. Therefore any polynomial-time ρ -approximation algorithm for the lecture-hall assignment problem, applied to this constructed instance, must return a solution with $\pi = \sigma$, because every solution with $\pi \neq \sigma$ already exceeds ρ times the optimum. For such a returned solution, the lecture-hall objective is exactly the normalized QAP objective of the same permutation. Hence a polynomial-time ρ -approximation for the lecture-hall assignment problem yields a polynomial-time ρ -approximation for QAP.

Thus both exact hardness and approximation hardness follow from the same reduction family. \square

Corollary 2. *Unless $P = NP$, there is no polynomial-time constant-factor approximation algorithm for the lecture-hall assignment problem. The same inapproximability holds even when the input is restricted to a single-day instance with two consecutive time blocks, identical hall capacities, and full lecture-hall compatibility.*

Proof. Sahni and Gonzalez (1976) proved that, unless $P = NP$, no polynomial-time algorithm approximates the classical QAP within any constant factor. By Proposition 1, any polynomial-time ρ -approximation algorithm for the lecture-hall assignment problem would, for the same ρ , yield a polynomial-time ρ -approximation for QAP. The reduction in the proof of Proposition 1 uses only a single day, two consecutive time blocks, identical hall capacities, and full compatibility, so the inapproximability transfers verbatim to that restricted input class. \square

3. Models and algorithms

This section presents three complementary exact modeling views of the lecture-hall assignment problem. We begin with a natural mixed-integer quadratic programming formulation, then derive a compact mixed-integer linear programming reformulation, and finally present a constraint-programming formulation that captures the assignment structure directly. All three base formulations optimize the same combined objective from Section 2: the quadratic walking term weighted by shared enrollments and hall-to-hall distances, plus the linear assignment-penalty term $\sum_l p_{l,a_l}$. After these three base formulations, we describe several modeling improvements that can be layered on top of them, including valid capacity-dominance inequalities, biclique cuts, and safe compatibility preprocessing.

For notational convenience, let $A' \subseteq A$ denote the set of *active successor pairs*, that is, the immediate-successor pairs whose two lectures share at least one enrolled student:

$$A' = \{(l_1, l_2) \in A : c_{l_1, l_2} > 0\}.$$

Pairs in $A \setminus A'$ carry zero shared enrollment and therefore contribute nothing to the walking term of the objective, so throughout the formulations below the quadratic walking cost and the auxiliary distance variables are indexed over A' rather than A .

3.1. Mixed-Integer Quadratic Programming Formulation

We first observe that the problem is separable by day. Let D denote the set of teaching days. For each $d \in D$, let $L_d \subseteq L$ be the set of lectures scheduled on day d , and let $A_d \subseteq A$ be the set of successor pairs whose two lectures both belong to day d . Because temporal-overlap constraints involve only lectures that occur on the same day, and because an immediate-successor pair must also lie within a single day, both the objective and feasibility constraints decompose across days. Therefore, the weekly problem can be written as the sum of independent daily subproblems,

$$\min \sum_{d \in D} f_d(x^d), \quad (1)$$

where each f_d is the objective induced by the lectures and successor pairs of day d , and x^d denotes the hall-assignment decisions for that day. Solving the overall weekly problem is therefore equivalent to solving one independent lecture-hall assignment problem for each day and combining the resulting assignments. In the remainder of Section 3 we work on a single-day instance and suppress the day index: L denotes the lectures of that day, A (and the derived A') the induced successor pairs, $G = (L, E)$ the interval overlap graph, and \mathcal{C} the family of maximal cliques of G . The full-week problem is recovered by solving one such instance per day.

A natural mixed-integer quadratic programming (MIQP) formulation is then as follows. For each lecture $l \in L$ and each compatible hall $h \in H(l)$, let

$$x_{lh} = \begin{cases} 1 & \text{if lecture } l \text{ is assigned to hall } h, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Recall that s_l is the number of students attending lecture l , u_h is the capacity of hall h , and $H(l) \subseteq H$ is the compatibility set defined in Section 2 that respects both the capacity bound $u_h \geq s_l$ and the mandatory-feature inclusion $F_l^{\text{req}} \subseteq F_h$. The problem can then be written as

$$\min \sum_{(l_1, l_2) \in A'} c_{l_1, l_2} \sum_{h_1 \in H(l_1)} \sum_{h_2 \in H(l_2)} d_{h_1, h_2} x_{l_1 h_1} x_{l_2 h_2} + \sum_{l \in L} \sum_{h \in H(l)} p_{lh} x_{lh} \quad (3)$$

$$\text{s.t.} \quad \sum_{h \in H(l)} x_{lh} = 1, \quad \forall l \in L, \quad (4)$$

$$\sum_{l \in C: h \in H(l)} x_{lh} \leq 1, \quad \forall C \in \mathcal{C}, \forall h \in H, \quad (5)$$

$$x_{lh} \in \{0, 1\}, \quad \forall l \in L, \forall h \in H(l). \quad (6)$$

The objective has two parts. The first is quadratic: the walking cost of assigning a lecture to a hall depends on where the immediately succeeding lecture is assigned, and each contribution is scaled by the shared-enrollment weight c_{l_1, l_2} . The second is linear: each compatible lecture–hall pair (l, h) contributes its local assignment penalty p_{lh} whenever it is selected. The first set of constraints ensures that each lecture is assigned to exactly one compatible hall, and the second enforces non-overlap by requiring that, for every hall and every maximal clique of mutually overlapping lectures, at most one lecture in that clique may use the hall.

3.2. Mixed-Integer Programming Reformulation

A compact linearization can be obtained by introducing, for each successor pair $(l_1, l_2) \in A$ such that $c_{l_1, l_2} > 0$, one nonnegative continuous variable $z_{l_1 l_2}$ representing the realized hall-to-hall distance induced by that pair. The linear assignment-penalty term is already linear in the x variables

and so is transferred verbatim from (3). The problem can then be reformulated as the mixed-integer linear program

$$\min \quad \sum_{(l_1, l_2) \in A'} c_{l_1, l_2} z_{l_1 l_2} + \sum_{l \in L} \sum_{h \in H(l)} p_{lh} x_{lh} \quad (7)$$

s.t. Constraints (4) and (5)

$$z_{l_1 l_2} \geq d_{h_1, h_2} (x_{l_1 h_1} + x_{l_2 h_2} - 1), \quad \forall (l_1, l_2) \in A', h_1 \in H(l_1), h_2 \in H(l_2), \quad (8)$$

$$x_{lh} \in \{0, 1\}, \quad \forall l \in L, \forall h \in H(l), \quad (9)$$

$$z_{l_1 l_2} \geq 0, \quad \forall (l_1, l_2) \in A'. \quad (10)$$

This reformulation is exact. The assignment-penalty term is the same expression as in the MIQP model and therefore needs no further argument. For the quadratic walking term, since each lecture is assigned to exactly one hall, for every pair (l_1, l_2) there is exactly one compatible hall pair (h_1, h_2) for which $x_{l_1 h_1} = x_{l_2 h_2} = 1$; for that pair, constraint (8) yields

$$z_{l_1 l_2} \geq d_{h_1, h_2}.$$

For every other compatible hall pair $(h'_1, h'_2) \neq (h_1, h_2)$, at least one of $x_{l_1 h'_1}$ and $x_{l_2 h'_2}$ equals zero, so the bracket $x_{l_1 h'_1} + x_{l_2 h'_2} - 1$ is nonpositive; combined with $d_{h'_1, h'_2} \geq 0$, the right-hand side of (8) is then at most zero and the corresponding constraint is already implied by the nonnegativity bound $z_{l_1 l_2} \geq 0$. Therefore, because the objective minimizes $\sum c_{l_1, l_2} z_{l_1 l_2}$ with $c_{l_1, l_2} \geq 0$ and $z_{l_1 l_2} \geq 0$, the optimal value of $z_{l_1 l_2}$ is exactly the realized distance d_{h_1, h_2} associated with the halls assigned to l_1 and l_2 . Multiplying by c_{l_1, l_2} in the objective recovers the original walking-cost contribution. Restricting the z variables to pairs with $c_{l_1, l_2} > 0$ is sufficient, since pairs with zero shared enrollment do not contribute to the walking part of the objective, and the assignment-penalty part is independent of z .

Compared with the standard disaggregated linearization of QAP that introduces one variable for each tuple (l_1, l_2, h_1, h_2) , formulation (7)–(10) is substantially more compact in the number of additional variables. In addition, placing the shared-enrollment weight c_{l_1, l_2} in the objective rather than in the linking constraints yields a better-scaled coefficient matrix while preserving the exact same optimal assignment. However, this compactness does not necessarily imply a stronger linear-programming relaxation on the linking side. Strengthening inequalities for this compact reformulation are described in Subsection 3.4.

3.3. Constraint Programming Formulation

A CP formulation captures the assignment structure more directly by replacing the binary assignment variables with one integer decision variable a_l for each lecture $l \in L$, whose domain is restricted to the compatible set $H(l)$. Because CP-SAT operates natively on integer variables with finite domains, the walking distance and the per-lecture assignment penalty are not passed to the solver as symbolic expressions $d_{a_{l_1}, a_{l_2}}$ and p_{l, a_l} , but as auxiliary integer variables tied to the hall assignments through extensional (tuple-table) constraints. Specifically, for each lecture $l \in L$ introduce a nonnegative integer variable

$$q_l \in \{p_{lh} : h \in H(l)\}$$

that records the realized assignment penalty, and for each successor pair $(l_1, l_2) \in A'$ introduce a nonnegative integer variable

$$z_{l_1 l_2} \in \{d_{h_1, h_2} : h_1 \in H(l_1), h_2 \in H(l_2)\}$$

that records the realized hall-to-hall walking distance. The CP model is then

$$\min \sum_{(l_1, l_2) \in A'} c_{l_1, l_2} z_{l_1 l_2} + \sum_{l \in L} q_l \quad (11)$$

$$\text{s.t. } a_l \in H(l), \quad \forall l \in L, \quad (12)$$

$$\text{AllDifferent}(\{a_l : l \in C\}), \quad \forall C \in \mathcal{C}. \quad (13)$$

$$(a_l, q_l) \in \{(h, p_{lh}) : h \in H(l)\}, \quad \forall l \in L, \quad (14)$$

$$(a_{l_1}, a_{l_2}, z_{l_1 l_2}) \in \{(h_1, h_2, d_{h_1, h_2}) : h_1 \in H(l_1), h_2 \in H(l_2)\}, \quad \forall (l_1, l_2) \in A', \quad (15)$$

The objective (11) sums, over successor pairs with positive shared enrollment, the weight c_{l_1, l_2} times the realized walking distance $z_{l_1 l_2}$, plus the per-lecture assignment penalty q_l . Constraint (12) encodes the hard hall compatibility requirements through the domain of a_l , so no separate assignment equalities are needed. Constraint (13) enforces temporal feasibility: for every maximal clique C of mutually overlapping lectures, the lectures in C must take distinct hall values, so no hall is used simultaneously by two overlapping lectures. Constraints (14) and (15) are the tuple-table encodings of the relations $q_l = p_{l, a_l}$ and $z_{l_1 l_2} = d_{a_{l_1}, a_{l_2}}$, respectively: each hall (or compatible hall pair) contributes one tuple that pairs the hall configuration with its induced penalty or distance value, so the auxiliary variable is pinned by the choice of a as soon as the relevant hall domain values are fixed.

Our OR-Tools CP-SAT implementation posts both tuple-table relations through the **AddAllowedAssignments** API, which takes an ordered list of integer variables together with the explicit list of allowed tuples shown in (14) and (15). We prefer this explicit tabular encoding to an **AddElement**-based alternative for two reasons. First, **AddAllowedAssignments** directly represents the joint relation between a pair of index variables and the looked-up value in a single constraint, whereas **AddElement** indexes a fixed one-dimensional array by a single integer variable and therefore does not express the two-index distance lookup $z_{l_1 l_2} = d_{a_{l_1}, a_{l_2}}$ without auxiliary intermediate variables. Second, the tuple tables list only halls (or hall pairs) that belong to the compatibility sets, so after the preprocessing of Subsection 3.4 the distance table for pair (l_1, l_2) has size exactly $|H(l_1)| \cdot |H(l_2)|$, and no infeasible (h_1, h_2) combination is ever enumerated. As in the MIP reformulation of Subsection 3.2, it is enough to materialize $z_{l_1 l_2}$ only for pairs $(l_1, l_2) \in A'$, since pairs with zero shared enrollment do not contribute to the walking part of the objective.

3.4. Modeling improvements

Several enhancements can be layered on top of the three base formulations without changing the underlying problem. All three improvements presented below apply to each of the MIQP, MIP, and CP formulations; what differs is only the encoding, which adapts to each modeling view's natural primitives: linear inequalities in the MIP, bilinear inequalities in the MIQP, and auxiliary Boolean indicators or tuple-table propagators in the CP model.

Capacity-dominance cuts. A first general improvement combines the hall-capacity structure with the clique-overlap restriction. For each maximal clique $C \in \mathcal{C}$ and each capacity threshold $\kappa \geq 0$, define the sets

$$L_\kappa(C) = \{l \in C : s_l > \kappa\}, \quad H_\kappa = \{h \in H : u_h > \kappa\}.$$

Every lecture $l \in L_\kappa(C)$ has $H(l) \subseteq H_\kappa$ by the capacity definition, so all of its assignment mass is placed on halls in H_κ . Summing the clique-overlap inequalities (5) over $h \in H_\kappa$ and isolating the contribution of the large lectures yields the *capacity-dominance cut*

$$\sum_{l \in C \setminus L_\kappa(C)} \sum_{h \in H(l) \cap H_\kappa} x_{lh} \leq |H_\kappa| - |L_\kappa(C)|, \quad \forall C \in \mathcal{C}, \forall \kappa \geq 0. \quad (16)$$

Because $|L_\kappa(C)|$ and $|H_\kappa|$ depend only on the input data, the cut is an explicit linear inequality in the assignment variables whose right-hand side is a pre-computed nonnegative integer. It states that once the $|L_\kappa(C)|$ largest lectures of the clique have claimed their share of the $|H_\kappa|$ halls whose capacity exceeds κ , the remaining lectures in C may collectively occupy at most $|H_\kappa| - |L_\kappa(C)|$ of the large halls. Only threshold values κ that coincide with distinct hall capacities need to be instantiated, so the family contains at most $|\mathcal{C}| \cdot |\{u_h : h \in H\}|$ inequalities. The cut is a Chvátal–Gomory consequence of (4)–(5) and is therefore valid for every integer-feasible solution of the MIQP, MIP, and CP formulations alike. In the MIQP and MIP formulations it can be added directly as a linear inequality in the assignment variables. In the CP formulation it can be represented by Boolean indicators that record whether a lecture is assigned to a hall in H_κ , followed by the corresponding redundant upper bound on the sum of those indicators. If preprocessing is applied beforehand, the cut can be posted on the reduced compatibility sets without modification.

Biclique cuts. The next family of improvements strengthens the distance linking between successor pairs. We present it first for the compact MIP reformulation, and then extend it to the MIQP and CP formulations. Fix a pair $(l_1, l_2) \in A'$ and a pair of halls $h_1 \in H(l_1)$ and $h_2 \in H(l_2)$, let $S_1 \subseteq H(l_1)$ and $S_2 \subseteq H(l_2)$ such that $d_{h'_1, h'_2} \geq d_{h_1, h_2}$ for every $h'_1 \in S_1$ and $h'_2 \in S_2$. Then the following inequality is valid:

$$z_{l_1 l_2} \geq d_{h_1, h_2} \left(\sum_{h'_1 \in S_1} x_{l_1 h'_1} + \sum_{h'_2 \in S_2} x_{l_2 h'_2} - 1 \right), \quad \forall (l_1, l_2) \in A', h_1 \in H(l_1), h_2 \in H(l_2). \quad (17)$$

If the assigned hall of l_1 lies in S_1 and that of l_2 lies in S_2 , then both summations equal 1 and (17) forces $z_{l_1 l_2} \geq d_{h_1, h_2}$, which is correct since the realized distance is at least d_{h_1, h_2} . Otherwise the right-hand side is nonpositive and the cut is implied by the nonnegativity of $z_{l_1 l_2}$. The basic constraints (8) are the special case $S_1 = \{h_1\}$ and $S_2 = \{h_2\}$; taking larger bicliques yields strictly stronger cuts that can separate fractional solutions satisfying all pairwise constraints.

In practice we generate a family of patterns (S_1, S_2, δ) by the anchor-based procedure in Algorithm 1. For each anchor pair $(h_1, h_2) \in H(l_1) \times H(l_2)$ the algorithm sets the common threshold $\delta = d_{h_1, h_2}$, expands S_2 to every hall in $H(l_2)$ at distance at least δ from the anchor h_1 , and then expands S_1 to every hall in $H(l_1)$ at distance at least δ from every element of S_2 . The anchor itself always satisfies $h_1 \in S_1$ and $h_2 \in S_2$, so (17) evaluated at the returned triple is at least as strong as the base pair-distance constraint obtained from the same anchor. Distinct anchors often produce the same triple, which is why the generated patterns are deduplicated before any cut is posted. For every returned triple (S_1, S_2, δ) we add one inequality of the form (17) with d_{h_1, h_2} replaced by δ , so the total number of biclique cuts posted for a successor pair is bounded by $|H(l_1)| \cdot |H(l_2)|$ and in practice is substantially smaller because of the deduplication. In the MIQP formulation, where the auxiliary distance variables $z_{l_1 l_2}$ are absent, the same biclique pattern (S_1, S_2, δ) gives the bilinear valid inequality

$$\sum_{h'_1 \in S_1} \sum_{h'_2 \in S_2} d_{h'_1, h'_2} x_{l_1 h'_1} x_{l_2 h'_2} \geq \delta \left(\sum_{h'_1 \in S_1} x_{l_1 h'_1} + \sum_{h'_2 \in S_2} x_{l_2 h'_2} - 1 \right), \quad (18)$$

which is posted directly as a quadratic constraint: when the assigned halls of l_1 and l_2 both lie in S_1 and S_2 , respectively, the left-hand side equals the realized bilinear distance contribution of the pair and is at least δ by construction of the pattern; otherwise the right-hand side is nonpositive and the cut is implied by nonnegativity of the bilinear terms. In the CP model of Subsection 3.3, the same logic is translated into propagation constraints enforcing lower bounds on the auxiliary variables $z_{l_1 l_2}$ whenever the two assigned halls fall into the corresponding threshold sets.

Input: compatibility sets $H(l_1), H(l_2)$, hall-to-hall distance matrix d .

Output: deduplicated set $\mathcal{P}(l_1, l_2)$ of patterns (S_1, S_2, δ) .

```

1:  $\mathcal{P}(l_1, l_2) \leftarrow \emptyset$ 
2: for each  $(h_1, h_2) \in H(l_1) \times H(l_2)$  do
3:    $\delta \leftarrow d_{h_1, h_2}$ 
4:    $S_2 \leftarrow \{h \in H(l_2) : d_{h_1, h} \geq \delta\}$ 
5:    $S_1 \leftarrow \{h \in H(l_1) : d_{h, h''} \geq \delta \text{ for every } h'' \in S_2\}$ 
6:    $\mathcal{P}(l_1, l_2) \leftarrow \mathcal{P}(l_1, l_2) \cup \{(S_1, S_2, \delta)\}$ 
7: end for
8: return  $\mathcal{P}(l_1, l_2)$ 
    
```

Compatibility preprocessing. A final improvement shrinks the compatibility sets $H(l)$ before solving any of the three formulations. Fix a target lecture $l' \in L$ and consider the auxiliary problem

$$\begin{aligned}
 U_{l'} = \max \quad & \sum_{h \in H(l')} u_h x_{l'h} \\
 \text{s.t.} \quad & \sum_{h \in H(l)} x_{lh} = 1, & \forall l \in L, \\
 & \sum_{l \in \mathcal{C}: h \in H(l)} x_{lh} \leq 1, & \forall \mathcal{C} \in \mathcal{C}, \forall h \in H, \\
 & x_{lh} \in \{0, 1\}, & \forall l \in L, \forall h \in H(l).
 \end{aligned}$$

This model keeps only the hard feasibility constraints and maximizes the capacity assigned to the single lecture l' . If $U_{l'}$ denotes its optimal value, then every hall $h \in H(l')$ with $u_h > U_{l'}$ may be removed from $H(l')$. Indeed, suppose that some globally feasible assignment uses lecture l' in a hall \hat{h} with $u_{\hat{h}} > U_{l'}$. That same assignment satisfies the preprocessing constraints above and attains objective value at least $u_{\hat{h}}$, contradicting the definition of $U_{l'}$. Therefore the reduced set

$$\widehat{H}(l') = \{h \in H(l') : u_h \leq U_{l'}\}$$

preserves all feasible solutions of the original problem. Applying this optimization once to every lecture yields one valid global reduction pass for the MIQP, the compact MIP, and the CP model alike. Moreover, the pass can be repeated on the reduced compatibility sets. The iteration is monotone in the following sense: removing a hall from $H(l'')$ for any lecture l'' can only shrink the feasible region of the auxiliary problem for every other target lecture l' , since one of the candidate assignments for l'' simply becomes unavailable. Hence $U_{l'}$ is non-increasing from one round to the next, and consequently $\widehat{H}(l')$ is non-increasing as well. Each round therefore preserves the feasible set of the preceding reduced model, the sequence of compatibility sets shrinks monotonically, and the iteration terminates at a fixed point after at most $\sum_{l \in L} |H(l)|$ rounds. In the MIQP and MIP formulations, this directly removes assignment variables and all associated overlap and linking constraints. In the CP model, it shrinks the domains of the assignment variables and reduces the size of the distance tables or element encodings. In practice, the preprocessing used in our implementation is exactly this fixed-point procedure.

One may also use a cheaper lightweight version. Let

$$L'(l') = \{l'\} \cup \{l \in L : l \text{ overlaps } l' \text{ in time}\},$$

and solve the same maximization problem with $L'(l')$ replacing L and with the overlap constraints restricted accordingly. Let its optimal value be $\widetilde{U}_{l'}$. To see that this is a valid relaxation, note that

any feasible solution x of the full auxiliary problem, when projected to the variables indexed by $L'(l')$, still satisfies every assignment constraint of $L'(l')$ and every overlap constraint restricted to $L'(l')$: dropping the nonnegative terms contributed by lectures outside $L'(l')$ from the left-hand side of a ≤ 1 inequality can only weaken it. Since the lightweight objective depends only on $x_{l',h}$ and the projection preserves its value,

$$\tilde{U}_{l'} \geq U_{l'}.$$

Consequently, deleting from $H(l')$ only the halls with capacity larger than $\tilde{U}_{l'}$ is still safe: it may leave more halls than the full preprocessing, but it cannot remove a hall that is needed by a globally feasible solution. As in the full version, one may apply the lightweight reduction repeatedly until a fixed point is reached. The same monotonicity argument carries over: shrinking $H(l'')$ for any lecture l'' can only tighten the lightweight auxiliary problem for l' in the next round, so $\tilde{U}_{l'}$ is non-increasing across rounds and the iteration terminates at a fixed point.

3.5. SameRoom and SameAttendees side constraints

Capacity limits, mandatory hall features, and soft per-assignment preferences are already part of the base model through the compatibility set $H(l)$ and the penalty coefficients p_{lh} defined in Section 2. However, the timetabling literature also considers pairwise side constraints that couple the hall choices of two different lectures, and these cannot be encoded through $H(l)$ or p_{lh} alone. In this subsection we show how the two most common such constraints from the ITC 2019 competition model (Müller et al., 2025), *SameRoom* and *SameAttendees*, are accommodated by the three formulations of Subsections 3.1–3.3, in both their hard and soft forms.

The *SameRoom* constraint requires two lectures, scheduled at non-overlapping times, to be assigned to the same hall. It can be formulated either as a hard constraint that must be satisfied by every feasible solution or as a soft constraint with an associated violation penalty. Let C_1^{hard} be the set of lecture pairs for which a hard SameRoom constraint is imposed. Then the following MIP and MIQP constraints express it:

$$x_{l_1,h} = x_{l_2,h} \quad \forall (l_1, l_2) \in C_1^{hard}, h \in H(l_1) \cap H(l_2), \quad (19)$$

and for the CP formulation it is even more straightforward:

$$a_{l_1} = a_{l_2} \quad \forall (l_1, l_2) \in C_1^{hard} \quad (20)$$

In the hard case, the compatible hall sets of the two lectures can be reduced to the intersection of their original feasible sets, that is,

$$H'(l_1) = H'(l_2) = H(l_1) \cap H(l_2).$$

Alternatively, suppose that SameRoom is a soft constraint for a pair of lectures l_1, l_2 , with penalty π_{l_1, l_2} if the two lectures are assigned to different halls. Let C_1^{soft} be the set of such pairs, and for each pair introduce a binary variable v_{l_1, l_2} that equals 1 when the constraint is violated. A linear encoding for the MIP formulation is

$$v_{l_1, l_2} \geq x_{l_1, h} - x_{l_2, h} \quad \forall (l_1, l_2) \in C_1^{soft}, h \in H(l_1) \cap H(l_2), \quad (21)$$

$$v_{l_1, l_2} \geq x_{l_2, h} - x_{l_1, h} \quad \forall (l_1, l_2) \in C_1^{soft}, h \in H(l_1) \cap H(l_2), \quad (22)$$

$$v_{l_1, l_2} \geq x_{l_1, h} \quad \forall (l_1, l_2) \in C_1^{soft}, h \in H(l_1) \setminus H(l_2), \quad (23)$$

$$v_{l_1, l_2} \geq x_{l_2, h} \quad \forall (l_1, l_2) \in C_1^{soft}, h \in H(l_2) \setminus H(l_1). \quad (24)$$

The corresponding penalty term

$$\sum_{(l_1, l_2) \in C_1^{soft}} \pi_{l_1, l_2} v_{l_1, l_2}$$

is then added to the objective function. In the MIQP formulation, one may avoid the auxiliary variables entirely and add the direct bilinear penalty

$$\sum_{(l_1, l_2) \in C_1^{soft}} \pi_{l_1, l_2} \left(1 - \sum_{h \in H(l_1) \cap H(l_2)} x_{l_1 h} x_{l_2 h} \right). \quad (25)$$

Because each lecture is assigned to exactly one hall, the inner sum equals 1 if and only if the two lectures use the same compatible hall, and equals 0 otherwise, so this term is exactly the SameRoom violation penalty. Implementation of the soft SameRoom constraint in the CP model is more straightforward: one may simply add the term

$$\sum_{(l_1, l_2) \in C_1^{soft}} \pi_{l_1, l_2} [a_{l_1} \neq a_{l_2}] \quad (26)$$

using a reification construct that evaluates to 1 when the logical condition is true and to 0 otherwise.

An important caveat is that SameRoom constraints, in both their hard and soft forms, are typically defined over the full weekly timetable. For example, a lecture given on Monday may be required to use the same hall as a related lecture given on Wednesday. If such cross-day constraints are enforced, the problem is no longer decomposable by day, and one must solve at least the whole week, and possibly the whole term, simultaneously.

Another common timetabling side constraint that is also relevant here, albeit in a more restricted sense, is *SameAttendees* (Müller et al., 2025). In the timetabling model, this constraint limits the travel time between two lectures that may share students. In our setting, where event times are fixed in advance, it becomes a restriction on the admissible hall pairs for two given lectures. Let C_2^{hard} be the set of lecture pairs with a hard SameAttendees constraint, and let

$$F_{l_1, l_2} \subseteq H(l_1) \times H(l_2)$$

denote the set of forbidden hall pairs for a given pair (l_1, l_2) . If l_1 ends before l_2 starts, then $(h_1, h_2) \in F_{l_1, l_2}$ whenever

$$d_{h_1, h_2} > \text{start}(l_2) - \text{end}(l_1).$$

If l_2 ends before l_1 starts, then $(h_1, h_2) \in F_{l_1, l_2}$ whenever

$$d_{h_2, h_1} > \text{start}(l_1) - \text{end}(l_2).$$

If the two lectures overlap in time, then every hall pair is forbidden. The hard constraint is therefore expressed in the MIP and MIQP models by

$$x_{l_1, h_1} + x_{l_2, h_2} \leq 1 \quad \forall (l_1, l_2) \in C_2^{hard}, (h_1, h_2) \in F_{l_1, l_2}. \quad (27)$$

In the CP model, one may equivalently forbid the assignments in F_{l_1, l_2} directly.

Now let C_2^{soft} be the set of soft SameAttendees pairs, with penalty ρ_{l_1, l_2} for violating pair (l_1, l_2) . Introduce a binary variable u_{l_1, l_2} that equals 1 when the chosen hall pair belongs to F_{l_1, l_2} . Then the MIP model can enforce

$$u_{l_1, l_2} \geq x_{l_1, h_1} + x_{l_2, h_2} - 1 \quad \forall (l_1, l_2) \in C_2^{soft}, (h_1, h_2) \in F_{l_1, l_2}, \quad (28)$$

and add the penalty term

$$\sum_{(l_1, l_2) \in C_2^{soft}} \rho_{l_1, l_2} u_{l_1, l_2}$$

to the objective. In the MIQP formulation, the same penalty can be written directly as

$$\sum_{(l_1, l_2) \in C_2^{soft}} \rho_{l_1, l_2} \sum_{(h_1, h_2) \in F_{l_1, l_2}} x_{l_1, h_1} x_{l_2, h_2}. \quad (29)$$

Because exactly one hall pair is selected for (l_1, l_2) , the inner sum is a binary indicator of whether the chosen pair belongs to F_{l_1, l_2} . This avoids both auxiliary variables and the accompanying linking inequalities in the MIQP model. In the CP model, the same logic can be implemented by a reified table constraint that links the chosen hall pair to a binary violation indicator.

In the MIP model, the pairwise formulations above can be strengthened using another set biclique constraints. Let $S_1 \subseteq H(l_1)$ and $S_2 \subseteq H(l_2)$ be such that

$$S_1 \times S_2 \subseteq F_{l_1, l_2}.$$

Then every assignment that places l_1 in a hall from S_1 and l_2 in a hall from S_2 necessarily violates SameAttendees. Hence the following aggregated inequalities are valid:

$$\sum_{h_1 \in S_1} x_{l_1, h_1} + \sum_{h_2 \in S_2} x_{l_2, h_2} \leq 1 \quad \forall (l_1, l_2) \in C_2^{hard}, S_1 \times S_2 \subseteq F_{l_1, l_2}, \quad (30)$$

and

$$u_{l_1, l_2} \geq \sum_{h_1 \in S_1} x_{l_1, h_1} + \sum_{h_2 \in S_2} x_{l_2, h_2} - 1 \quad \forall (l_1, l_2) \in C_2^{soft}, S_1 \times S_2 \subseteq F_{l_1, l_2}. \quad (31)$$

These inequalities dominate the basic pairwise constraints whenever $|S_1| > 1$ or $|S_2| > 1$. In the implementation, the extended cut mode constructs such sets by starting from one anchor hall of l_1 , collecting all halls of l_2 that are forbidden with that anchor, and then enlarging the l_1 -side set to every hall that is incompatible with the entire resulting set on the l_2 side. This is directly analogous in spirit to the extended distance cuts used for the quadratic walking term. These aggregated inequalities are useful for the linearized MIP and for the CP reified encoding; in the MIQP formulation, the soft SameAttendees penalties are represented directly by bilinear objective terms instead. In our CP implementation, the hard and soft SameAttendees constraints are encoded natively through forbidden-pair and reified table propagators over the domains of a_{l_1} and a_{l_2} , so the CP-SAT engine already performs full arc-consistent propagation on each pair; the biclique inequalities (30) and (31) are therefore posted only in the MIP model and are not replicated as additional CP constraints, since they would be redundant with the native table propagation.

Table 1 summarizes, for each of the four hard/soft variants of the two side constraints, the equation number that encodes it in each of the three formulations, giving the reader a single-glance map into the preceding pages. The two CP *SameAttendees* cells reference the native table constructs of the CP-SAT model rather than a numbered inequality, because those requirements are propagated directly on the domains of the hall-assignment variables a_i ; the MIP biclique strengthenings (30) and (31) are redundant under that native propagation and are therefore posted only in the MIP model.

Hard SameRoom and SameAttendees constraints can also be incorporated into the compatibility preprocessing step described in Subsection 3.4. For a hard SameRoom pair $(l_1, l_2) \in C_1^{hard}$, both compatibility sets may simply be replaced by their intersection $H(l_1) \cap H(l_2)$, after which the two lectures behave, for the purposes of the auxiliary preprocessing problem, as a single merged activity. For a hard SameAttendees pair $(l_1, l_2) \in C_2^{hard}$, one removes from $H(l_2)$ every hall h_2 such that

Table 1: Equations that encode each hard/soft side constraint in each of the three formulations.

Side constraint	MIQP	MIP	CP
SameRoom, hard	(19)	(19)	(20)
SameRoom, soft	(25)	(21)–(24)	(26)
SameAttendees, hard	(27)	(27); strengthened by (30)	forbidden-pair table on (a_{l_1}, a_{l_2})
SameAttendees, soft	(29)	(28); strengthened by (31)	reified table on (a_{l_1}, a_{l_2})

$\{h_1 \in H(l_1) : (h_1, h_2) \in F_{l_1, l_2}\} = H(l_1)$, i.e., every hall of l_2 that is forbidden against every remaining option of l_1 , and symmetrically shrinks $H(l_1)$. These reductions are safe, no feasible assignment is eliminated and they can be applied before or interleaved with the fixed-point capacity-preprocessing loop of Subsection 3.4, where they typically accelerate convergence by forcing additional overlap-implied eliminations.

4. Numerical experiments

This section reports the numerical evaluation of the proposed formulations and modeling improvements on the real-world benchmark. Subsection 4.1 describes how the daily instances are extracted from the ITC 2019 and Lancaster 2023 data sources. Subsection 4.2 then summarizes the 300-second computational campaign, compares the MIQP, MIP, and CP formulations, and studies the contribution of biclique strengthening, capacity-dominance cuts, and compatibility preprocessing. A detailed account of the results obtained for each instance by each formulation is presented in Appendix Appendix A.

The MIQP and MIP formulations were implemented in Python using Gurobi 13.0, and the CP formulation was implemented with Google OR-Tools CP-SAT, version 9.12. The same Python code base handles data preparation, preprocessing, and cut generation, and the experimental campaigns were executed by shell scripts. For reproducibility, the code, its documentation and detailed result files are available at <https://github.com/ta169/lecture-halls>. All experiments were run on an AMD Ryzen 9 5950X with 128GB of RAM under Ubuntu 22.04.

4.1. Test instances

Our real-world benchmark is derived from the ITC 2019 university course timetabling repository (Müller et al., 2025) and from a 2023 institutional dataset from Lancaster University, kindly provided by Matthew Davison, the first author of Davison et al. (2025). In both sources we retain the original hall set, hall capacities, and hall-to-hall distances, and transform a weekly timetable into independent daily lecture-hall assignment instances because the optimization model has no cross-day coupling constraints. For each source we first select the most loaded teaching week, where load is measured by the number of active hall-requiring lectures after the relevant preprocessing steps. We then retain only the five most loaded days of that week, which in the present data coincide with the weekday pattern, and discard weekend activity.

The ITC 2019 repository contains 36 real-world timetabling instances, namely 30 competition instances and 6 test instances (Müller et al., 2025). For the main large-scale experiments we use the five largest competition instances that simultaneously contain a non-trivial hall-distance matrix and explicit student-enrollment data. For each selected ITC source we use a public timetable posted in the ITC 2019 repository as the fixed weekly schedule from which daily hall-assignment instances are extracted. After selecting the peak week, we keep only its five busiest days and drop the weekend, so each ITC source contributes five daily instances. According to the official competition results posted on the ITC 2019 website (Müller et al., 2020), the winning submissions for the selected instances were

submitted by Dennis S. Holm, Rasmus Ø. Mikkelsen, Matias Sørensen, and Thomas R. Stidsen for `muni-pdf-spr16c`, `pu-proj-fal19`, `muni-pdfx-fal17`, and `pu-d9-fal19`, and by Karim Er-rhaimini for `agh-fal17`. Both winning approaches are documented in solver papers published on the contest website (Holm et al., 2020; Er-rhaimini, 2020). Table 2 summarizes the selected source instances and the resulting benchmark sizes.

For Lancaster University, we use the actual 2023 timetable and anonymized student-registration data used by the institution. We are grateful to Matthew Davison for making these data available. The Lancaster XML contains *SameClass* groups, that is, sets of lecture identifiers that are administratively distinct but correspond to the same teaching activity and must therefore share the same time and hall. Our preprocessing pipeline contracts each *SameClass* component before any hall-assignment instance is created. It then identifies the two main teaching terms and selects the most loaded week within each term, which in the present data are weeks 13 and 28. For each selected term week we again retain only the five most loaded days and discard weekend activity, so the Lancaster source contributes ten daily instances overall. A small number of registrations that violate the single-lecture-at-a-time assumption of Section 2 are removed from one of the two overlapping lectures, so that the cleaned data satisfy the assumption and the induced student-movement relation is well defined for the quadratic objective. After this cleaning, the hard and soft side constraints are inherited from the XML through the merged components; in the resulting Lancaster instances there are hard and soft *SameAttendees* constraints but no remaining *SameRoom* constraints.

Three aspects of the Lancaster data pipeline deserve explicit characterization, since they affect the construction of the inputs d_{h_1, h_2} and c_{l_1, l_2} used in the optimization models. First, the campus geometry is not supplied as a set of two-dimensional coordinates. Instead, Davison et al. (2025) encode the Lancaster campus as a sparse, symmetric travel-time matrix embedded directly in the ITC 2019 XML format (Müller et al., 2025). Each entry d_{h_1, h_2} is a nonnegative integer representing the number of five-minute time slots required to walk from hall h_1 to hall h_2 ; the granularity arises from the XML slot resolution of 288 slots per day, i.e., $1440/288 = 5$ minutes per slot. The 660 halls produce six distinct positive travel values $d_{h_1, h_2} \in \{1, 2, 3, 4, 5, 7\}$ (corresponding to 5, 10, 15, 20, 25, and 35 minutes of walking time), with the most common values being 1 and 2. Pairs of halls for which no `<travel>` entry appears in the XML are treated as co-located ($d_{h_1, h_2} = 0$); no hall features are defined, so the compatibility sets $H(l)$ are driven entirely by hall capacities.

Second, instead of using a published timetable solution as a fixed schedule (the approach used for the ITC instances), the Lancaster pipeline reads the zero-penalty time assignment embedded in the XML class definitions, which records the schedule that was actually used by the institution. Since each class has exactly one zero-penalty time option, the timetable is unique and deterministic: the *SameClass*-contracted merged components inherit their day, start, and duration fields directly from this institutional assignment. Third, the shared-enrollment weight c_{l_1, l_2} is constructed from individual student registrations, which are listed explicitly in the XML. Because many Lancaster courses offer multiple teaching configurations, the pipeline performs a greedy student-level assignment: for each student, courses are sorted by the number of feasible weekly-meeting options and each course is assigned to the most active feasible configuration that does not conflict with the student’s already-scheduled components and does not exceed the hall capacity implied by the merged-component’s assigned room. After this greedy pass, a student contributes to c_{l_1, l_2} if they attend both lectures l_1 and l_2 and the two lectures are immediate successors on the same day, that is, lecture l_2 starts within a short-break gap (automatically inferred from the most common positive inter-lecture gap in the data) after lecture l_1 ends.

Overall, the benchmark contains 35 daily instances: 25 from the five selected ITC source XML files and 10 from Lancaster. The ITC subset combines five large competition benchmarks, whereas

Table 2: Source instances underlying the real-world benchmark after retaining the five working days of the selected weeks, sorted in ascending order by number of halls. The column $|d|$ is the number of distinct values in the hall-to-hall distance matrix (including zero for self-distance and co-located pairs). The five ITC rows yield 25 daily lecture-hall assignment instances in total; the Lancaster row summarizes the two main 2023 teaching terms and yields 10 additional daily instances. A student is counted once if they appear in at least one retained day of the corresponding source, whereas registrations are summed over all retained days.

Source XML	Description	Halls	$ d $	Sel. wk(s)	Lect./day	Students	Regs.
<code>muni-pdf-spr16c</code>	Masaryk, Spring 2016	70	9	3	159–222	1,715	9,869
<code>muni-pdfx-fal17</code>	Masaryk, Fall 2017	86	9	12	197–388	5,083	38,256
<code>pu-d9-fal19</code>	Purdue, Fall 2019	224	10	2	846–977	35,014	286,385
<code>agh-fal17</code>	AGH UST, Fall 2017	327	12	4	537–558	6,192	78,854
<code>lancs-yr23</code>	Lancaster, 2023 T1–T2	660	7	13, 28	306–597	13,274	196,173
<code>pu-proj-fal19</code>	Purdue, Fall 2019	770	13	6	2,208–2,562	38,248	437,517

the Lancaster subset contributes a much larger hall inventory and a different data-generation pipeline based on cleaned institutional registrations. Together, these sources provide a useful test bed for assessing the robustness of the MIQP, MIP, and CP formulations across materially different levels of hall abundance, timetable density, and student-flow structure.

It should be noted that both data sources have previously appeared in course timetabling studies, but the corresponding literature is not directly comparable with our results because those studies optimize broader timetabling or hall-allocation objectives and do not include the quadratic walking term that is central here. Although one can retrospectively compute student walking distances for a published timetable, it would not be meaningful to compare our optimal or near-optimal solutions against solutions produced for a different objective function.

4.2. Analysis of numerical results

To evaluate the three formulations, we solved each of the 35 instances from Subsection 4.1 with MIQP, MIP, and CP under all eight combinations of the three modeling improvements from Subsection 3.4. Because the full compatibility preprocessing requires solving an optimization subproblem for every lecture and yields diminishing returns over the cheaper lightweight version, the experimental campaign omits the full setting and evaluates preprocessing strictly as a binary switch between none and the lightweight reduction. This yields 840 exact runs, each with a 300-second time limit. In addition, we solved the root relaxation of the linearized MIP under the same eight combinations, which adds 280 shorter runs and isolates the effect of preprocessing and cutting planes on the linear relaxation.

For each instance, let LB^* denote the largest lower bound obtained by any method in either campaign, including the dedicated root-relaxation runs. Throughout this section we report optimality gaps in the conventional conservative form $(z - \ell)/\ell$, where z is the incumbent value and ℓ is the lower bound used for the comparison. Accordingly, we define the *best-known gap* as $(z - LB^*)/LB^*$, and the *solver-certified gap* of a single run as $(z - \ell_{\text{run}})/\ell_{\text{run}}$, where ℓ_{run} is the final lower bound proven by that run. The solver-certified gap therefore differs from the best-known gap in that it is computed against a per-run lower bound rather than the global LB^* . Since LB^* is positive on every instance in our benchmark, the best-known gap is always well defined; the per-run gap is also well defined in practice, because every exact run produced a strictly positive lower bound. Because every one of the 840 exact runs produced an incumbent solution, Table 3 reports only the number of optimal solves, the mean solver-certified gap, the mean best-known gap, and the median solver runtime. We use the median runtime because the arithmetic mean is distorted by the 300-second censoring on unsolved instances; for consistency, the 280 root-relaxation runs summarized later in Table 4 also report median solver times under the same convention.

Table 3: Representative performance of the three formulations under a 300-second time limit. “Mean proved gap” and “Mean best-known gap” are means of the solver-certified and best-known optimality gaps, respectively, over the instances in each row. “Max proved gap” is the worst-case solver-certified gap across those instances. All rows use light compatibility preprocessing and no capacity-dominance cuts; biclique strengthening is enabled except for the ITC-MIQP row, where the variant without biclique strengthening produces better incumbents on the hardest Purdue instances.

Source	Formulation	Optimal	Mean proved gap (%)	Mean best-known gap (%)	Max proved gap (%)	Median time (s)
ITC 2019	MIQP	17/25	2.51	0.14	21.45	0.71
ITC 2019	MIP	23/25	0.09	0.09	1.11	0.08
ITC 2019	CP	16/25	18.10	3.70	126.94	45.03
Lancaster	MIQP	10/10	0.00	0.00	0.00	27.92
Lancaster	MIP	10/10	0.00	0.00	0.00	0.60
Lancaster	CP	10/10	0.00	0.00	0.00	45.83

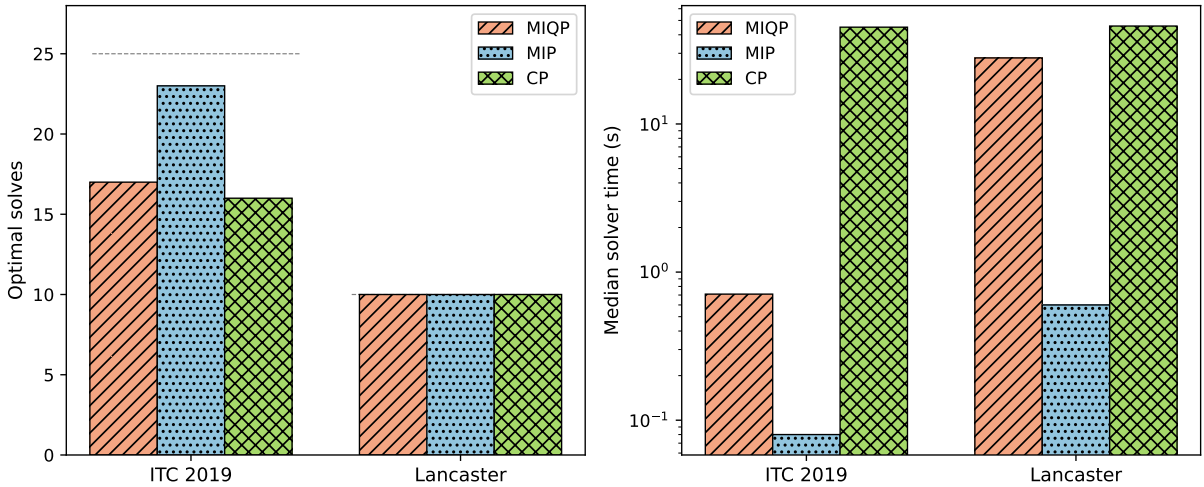


Figure 1: Optimal solve counts and median solver runtimes for the representative configurations reported in Table 3. The runtime panel uses a logarithmic scale.

For the overview in Table 3, we use a representative light-preprocessing, no-capacity-dominance block. Biclique strengthening is enabled for the MIP, CP, and Lancaster-MIQP rows. For ITC-MIQP, however, the variant without biclique strengthening produces clearly better incumbents on the hardest Purdue instances where Gurobi’s nonlinear branch-and-bound sometimes fails to return any incumbent under the stronger cuts within the 300-second limit, so that row uses the weaker setting instead. Figure 1 visualizes the same comparison.

Two broad conclusions are clear from Table 3, Figure 1, and the Dolan–Moré performance profile in Figure 2. First, the compact linearized MIP is the most robust exact formulation on this benchmark. On the 25 ITC-derived instances it solves 23 to proven optimality within the 300-second limit, compared with 17 for the MIQP model and 16 for CP-SAT. On the 10 Lancaster instances all three formulations solve every instance, but the MIP model is by far the fastest: its median solver runtime is about 0.6 seconds, compared with roughly 28 seconds for MIQP and 46 seconds for CP-SAT. This dominance is clearly visualized by the MIP curve strictly dominating both MIQP and CP-SAT in the performance profile.

Second, all three methods obtain incumbents throughout the campaign: every one of the 840 exact runs returns a feasible solution. The real difference lies in proof of optimality and in the quality of the final incumbents. This is reflected in the gap columns. On the harder ITC days, CP-SAT ends with a mean proved gap of 18.10% and a mean best-known gap of 3.70%, whereas the representative MIP block already reaches 0.09% and 0.09%, respectively. The representative MIQP

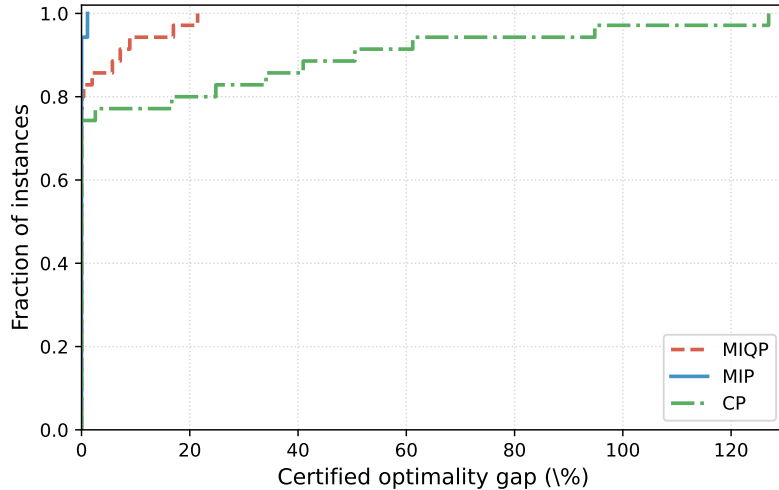


Figure 2: Empirical CDF of the certified optimality gap after 300 seconds for the MIQP, MIP, and CP formulations on the representative block (light preprocessing; capacity-dominance cuts off; biclique strengthening on, except ITC-MIQP which uses the weaker pairwise distance model). The x -axis spans 0% to 130% and each curve reports the fraction of the 35 benchmark instances with a certified gap at most x , reaching its ceiling at 1 once x exceeds the worst case or at $(n - k)/n$ if k of the n runs returned no incumbent.

block also stays close to the best available bounds, with a mean best-known gap of only 0.14%. The “Max proved gap” column reveals the tail behavior: the worst-case MIP gap is only 1.11% (on `pu-proj-fal19_week6_day3`), whereas the worst-case MIQP and CP-SAT gaps reach 21.45% and 126.94%, respectively, both on the hardest Purdue instances. These tails are visually formalized by the empirical CDF of certified gaps in Figure 2, where the MIP curve already hits 33/35 at 0% gap and captures 100% of instances by the 1.11% mark, whereas the MIQP and CP curves do not close 100% of instances until the 21.45% and 126.94% marks, respectively. Figure 3 stratifies these aggregate statistics by instance size: MIP and MIQP points cluster at or near 0% across the full size range, CP points coincide with them on every `agh`, `muni-pdf`, `muni-pdfx`, and Lancaster instance, and the only visible spread in either axis occurs on the largest `pu-d9` and `pu-proj` days, where CP best-known gaps fan out to between 1 and 41% and MIQP occasionally trails MIP by a percentage point or two. In other words, the Section 4 ranking between formulations is driven almost entirely by the largest dense instances, and the two formulation families are statistically indistinguishable on the remaining twenty-six benchmark days.

Figure 4 summarizes the 35 *best* solutions each formulation produces across the eight cut/preprocess settings: for each instance i and formulation f we take the smallest objective $z_{i,f}^*$ found by f on i and report its gap $(z_{i,f}^* - LB_i^*)/LB_i^*$ against the globally best known lower bound. This isolates the primal quality the user would observe in a production deployment that, for each (family, formulation) combination, simply runs the best configuration. The three distributions all have median 0%, and both MIP and MIQP have empty interquartile ranges, so their medians and 75th percentiles coincide at zero. MIP is slightly tighter than MIQP (mean 0.061% vs 0.088%; max 1.11% vs 1.59%; five vs six instances with a strictly positive gap), but the two are statistically indistinguishable. CP is noticeably more dispersed: mean 2.09%, standard deviation 6.46pp, max 27.66%, and nine out of 35 instances still carry a positive best-known gap. Every nonzero CP point sits on a `pu-d9` or `pu-proj` day; every ITC `agh`, `muni-pdf`, `muni-pdfx` day and every Lancaster day appears at the origin of every violin, confirming that the formulation ranking is determined entirely by the hardest dense instances and that on the remaining twenty-six days the three formulations are primal-equivalent.

The main driver of this performance difference is the biclique-style distance strengthening. Ta-

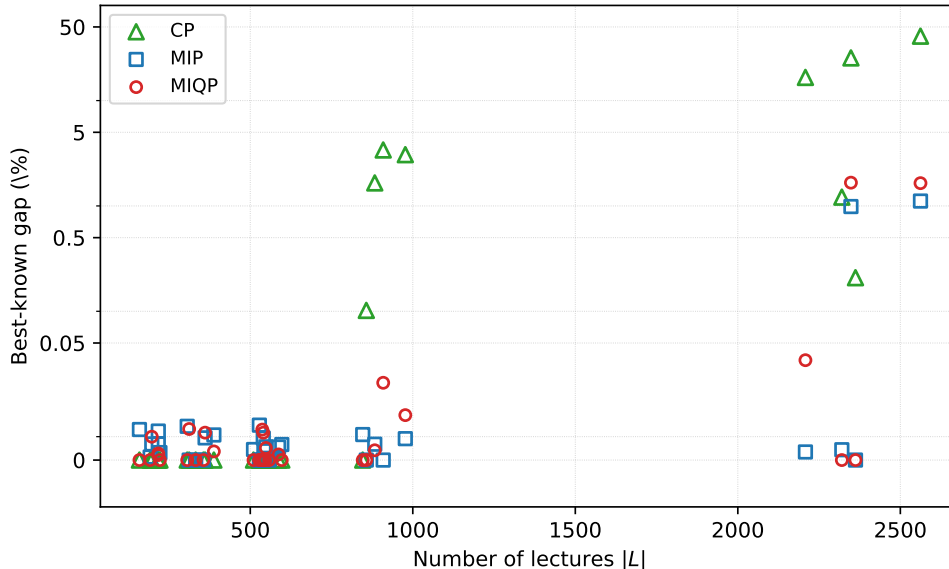


Figure 3: Best-known gap $bk\% = 100(z - LB^*)/LB^*$ after 300 seconds plotted against the number of lectures $|L|$ for the 35 representative instances. Each instance contributes three markers, one per formulation; the y -axis is shown on a symmetric-log scale with a linear band of width 0.05 around the origin, and MIQP and MIP markers are jittered horizontally by a small random amount to disambiguate overlapping zeros.

Table 4: Root-bound quality of the linearized model under light preprocessing and without capacity-dominance cuts. The shortfall is measured against the best-known lower bound LB^* . Root solver times are reported as medians over the instance set. “Weak” denotes the weaker pairwise distance model used as the baseline in the root experiments, and “Strong” denotes the biclique-enhanced distance model.

Source	Weak shortfall (%)	Strong shortfall (%)	Weak root time (s)	Strong root time (s)
ITC 2019	5.76	0.09	1.95	0.09
Lancaster	20.67	0.01	55.25	0.62

Table 4 and Figure 5 show its effect on the root relaxation of the linearized model under the same light-preprocessing, no-capacity-dominance setting. Here we report the *root shortfall*, namely the relative gap between the root lower bound and LB^* . On the ITC instances, the mean root shortfall drops from 5.76% for the weaker pairwise model to 0.09% for the biclique-enhanced model. On the Lancaster instances, the corresponding reduction is from 20.67% to 0.01%. The time needed to compute the root bound also decreases sharply: the median root solver time drops from about 1.9 to 0.09 seconds on ITC and from about 55 to 0.6 seconds on Lancaster.

Table 5 isolates each of the three levers. For the *linearized MIP* the biclique distance cuts are the decisive lever: enabling them triples the optimal count from 16 to 32–33 (out of 35), raises the best-known count from 20–21 to 33–35, and collapses the mean best-known gap from the 3–5% range to 0.06% regardless of the other two switches, with the mean provable gap dropping from above 11% to at most 0.08%. Preprocessing and the capacity-dominance cuts are essentially neutral on top of biclique: the four MIP rows with biclique on are statistically indistinguishable on every column. For the *MIQP* the picture is inverted. Without biclique, every MIQP combination reaches a mean best-known gap of 0.10% and a mean provable gap of 1.57–1.79% while returning an incumbent on every one of the 35 instances. Turning biclique on, by contrast, makes Gurobi’s nonlinear branch-and-bound drop the incumbent on three to five of the hardest Purdue instances in every one of the four biclique-on combinations, which leaves both the mean provable gap and the mean best-known gap undefined in the table (entries “–”); on the instances that do return an incumbent, # BK stays at 27–28 rather than improving. Preprocessing and the capacity-dominance cuts amplify rather than mitigate this degradation on MIQP, because they interact with the already fragile bilinear relaxation

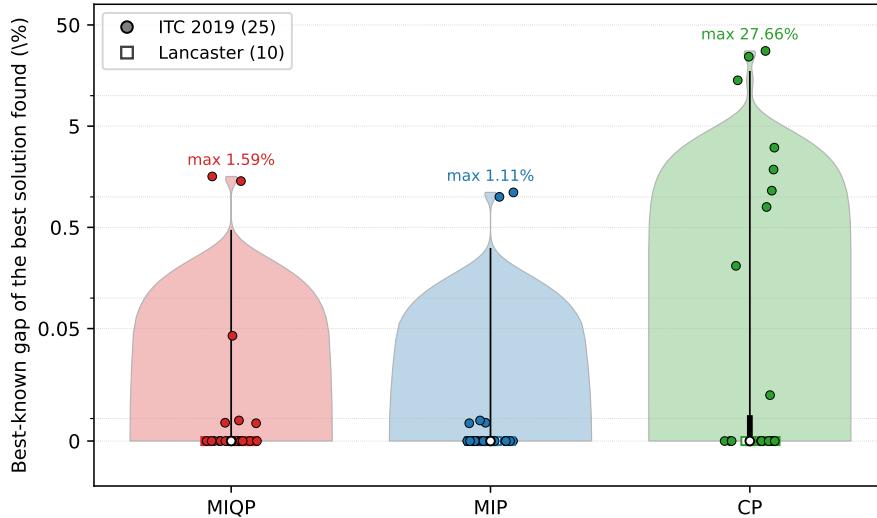


Figure 4: Distribution across the 35 benchmark instances of the best-known gap $\text{bk}\% = 100(z_{i,f}^* - LB_i^*)/LB_i^*$ of the best solution each formulation found, where $z_{i,f}^*$ is the smallest objective obtained by formulation f on instance i over the eight cut/preprocess settings and LB_i^* is the best known lower bound for instance i across every exact and root run. Filled circles mark ITC instances, open squares mark Lancaster instances; white dots mark medians, thick vertical bars mark interquartile ranges, and thin whiskers span the [5%, 95%] range. The y -axis uses a symmetric-log scale with a linear band of width 0.05 around the origin.

instead of helping it. For *CP-SAT* none of the three levers moves the numbers much: the best-known gap sits in a narrow 2.56–3.87% band, both the optimal count and the best-known count oscillate between 25 and 26, so *CP-SAT* is essentially indifferent to the modeling choices and benefits most from primal heuristics inside the solver rather than from the additional inequalities. The Lancaster instances are saturated by every *CP* and *MIQP* combination and by every *MIP* combination with biclique on, so the dispersion in Table 5 is driven almost entirely by the 25 ITC days, and in particular by the five largest Purdue instances. The practical upshot is that a production deployment should pair each formulation with its dominating combination, namely light preprocessing with biclique on and capacity cuts off for *MIP*, any biclique-off combination for *MIQP*, and any combination with the capacity cuts off for *CP-SAT*; these are the per-formulation best settings used in the representative block of Table 3 and in the dispersion summary of Figure 4.

The exact *MIP* results mirror the root-bound picture. On the ITC instances, the representative light-preprocessed *MIP* block improves from 16 optimal solves under the weaker distance model to 23 under the biclique-enhanced one, while the mean best-known gap drops from 7.14% to 0.09%. On Lancaster, the weak *MIP* times out on all ten instances within the 300-second limit, whereas the biclique-enhanced variant solves all ten to proven optimality in under a second of median runtime. The *CP* formulation also benefits from stronger distance reasoning on Lancaster, where its median solver runtime falls from about 75 to about 46 seconds, but its ITC improvements are much smaller. For the *MIQP* runs, biclique strengthening is clearly beneficial on Lancaster, where it lifts the solve count from 9/10 to 10/10 and cuts the median solver runtime from about 73 to 28 seconds, but is actively detrimental on the hardest ITC instances, where the nonlinear branch-and-bound under the stronger cuts sometimes fails to return any incumbent within 300 seconds and consequently inflates the mean best-known gap from 0.14% to 11.98% while leaving the number of proved-optimal solves unchanged at 17/25.

The other two modeling improvements are useful but clearly secondary in this campaign. Light compatibility preprocessing removes on average only 3.63% of compatibility entries on the ITC instances and 1.41% on the Lancaster instances, so large runtime changes should not be expected

Table 5: Per-(formulation, cut/preprocess-combination) statistics across the 35 benchmark instances under the 300-second time limit. “PP” is compatibility preprocessing (*light* or *none*); “Bic” toggles the biclique distance cuts; “Dominance” toggles the Chvátal–Gomory capacity-dominance cuts. “Mean best-known gap” averages $(z - LB^*)/LB^*$ and “Mean provable gap” averages $(z - LB)/LB$ with the run’s own lower bound LB . Both averages are replaced by “–” whenever at least one of the 35 instances returned *no incumbent* (solver proved a valid lower bound but produced no feasible solution inside the time limit, so the per-instance gap is undefined and the cross-instance mean would be misleading). All four MIQP biclique-on rows trigger this rule because Gurobi’s nonlinear branch-and-bound fails to return a feasible solution on three to five of the hardest Purdue days; every CP-SAT run, by contrast, returned both an incumbent and a positive lower bound on every instance, so the CP rows never take a “–”. “# Opt.” is the number of instances solved to proven optimality within the time limit; “# BK” is the number of instances on which the run tied the globally best incumbent objective found anywhere in the factorial.

Formulation	PP	Bic	Dominance	Mean best-known gap (%)	Mean provable gap (%)	# Opt.	# BK
MIQP	none	off	off	0.10	1.63	25	32
MIQP	none	off	on	0.10	1.59	26	30
MIQP	none	on	off	–	–	27	27
MIQP	none	on	on	–	–	27	27
MIQP	light	off	off	0.10	1.79	26	31
MIQP	light	off	on	0.10	1.57	26	30
MIQP	light	on	off	–	–	27	27
MIQP	light	on	on	–	–	28	28
MIP	none	off	off	3.14	11.04	16	21
MIP	none	off	on	3.47	11.43	16	20
MIP	none	on	off	0.06	0.07	33	33
MIP	none	on	on	0.06	0.08	32	33
MIP	light	off	off	5.13	13.36	16	21
MIP	light	off	on	3.67	11.59	16	21
MIP	light	on	off	0.06	0.06	33	35
MIP	light	on	on	0.06	0.06	32	34
CP	none	off	off	2.60	9.83	26	26
CP	none	off	on	3.46	13.65	25	25
CP	none	on	off	2.63	12.92	26	26
CP	none	on	on	3.87	17.39	26	26
CP	light	off	off	2.56	9.83	26	26
CP	light	off	on	2.84	11.77	25	25
CP	light	on	off	2.64	12.93	26	26
CP	light	on	on	3.70	16.49	26	26

from preprocessing alone. The saved runs are consistent with that expectation: preprocessing usually yields a modest speed-up, but it does not alter the qualitative ranking between formulations.

The capacity-dominance inequalities likewise have only a limited aggregate effect. To isolate their contribution we repeated the representative block of Table 3 with the cuts enabled, keeping all other settings fixed, and report the resulting deltas in Table 6. The cuts are essentially neutral on Lancaster: every instance was already solved to proven optimality without them, so neither the optimal count nor the best-known gap moves, and the median runtime shifts by at most 1.4 seconds in either direction. On ITC they are a net negative for the MIP and CP formulations. They flip `pu-d9-fal19-week2_day3` from an optimal solve in about 263 seconds to a MIP time-out, leaving the other MIP solves essentially unchanged, and they noticeably worsen the CP-SAT block: the mean solver-certified gap rises from 18.10% to 23.09%, the mean best-known gap from 3.70% to 5.18%, and the median CP-SAT runtime grows from 45 to 75 seconds, all without closing any additional instance. The ITC MIQP block is the closest to neutral: the cuts tighten the mean solver-certified gap from 2.51% to 2.19%, but the mean best-known gap ticks up by 0.01 percentage points and the solve count is unchanged, so the certified-gap improvement reflects a marginally tighter lower bound rather than a better incumbent. The cut generation itself is cheap, taking at most 65 milliseconds per run on the largest Purdue instances, so the observed slowdowns come from the interaction between the extra inequalities and the solvers’ branching, not from cut preparation.

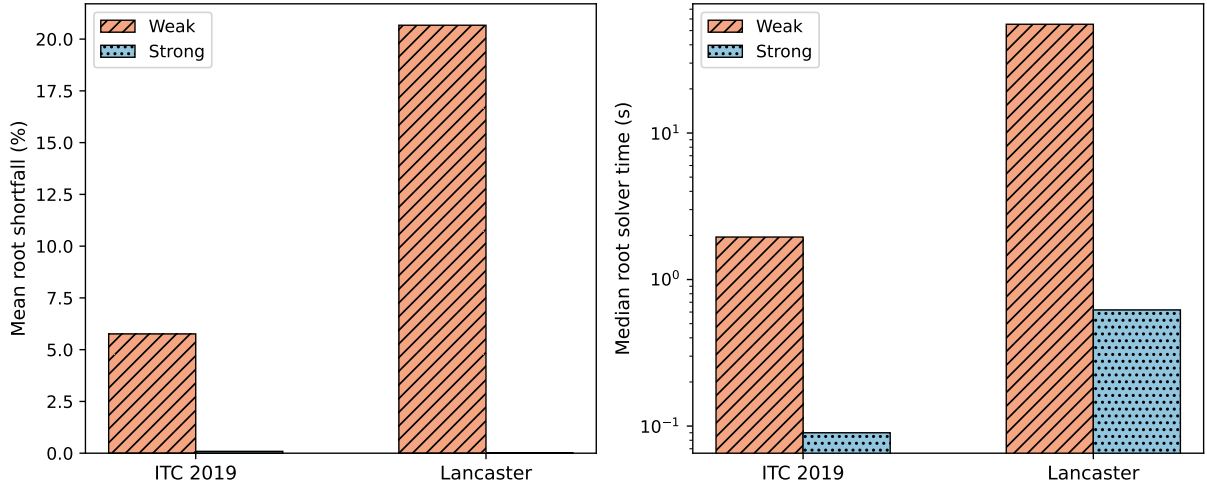


Figure 5: Mean root shortfalls and median root solver times for the weaker and strongest linearized distance models. The runtime panel uses a logarithmic scale, matching the convention of Figure 1.

Table 6: Ceteris paribus effect of enabling capacity-dominance cuts, measured as the difference between the representative block of Table 3 (cuts off) and the same block with cuts on. Positive Δ Optimal indicates more proved-optimal solves; positive Δ Median time indicates a slower median runtime. Units of the gap delta are percentage points.

Source	Formulation	Δ Optimal	Δ Mean best-known gap (pp)	Δ Median time (s)	Cut gen. time (s)
ITC 2019	MIQP	+0	+0.01	+0.00	0.061
ITC 2019	MIP	-1	+0.00	+0.01	0.065
ITC 2019	CP	+0	+1.48	+29.94	0.062
Lancaster	MIQP	+0	+0.00	-0.09	0.032
Lancaster	MIP	+0	+0.00	+0.02	0.032
Lancaster	CP	+0	+0.00	+1.44	0.032

A plausible explanation for this weak aggregate effect is that the compatibility sets $H(l)$ in both benchmarks are driven predominantly by non-capacity factors, such as required features, departmental or programme ownership of halls, and other institutional preferences, rather than by the capacity bound $u_h \geq s_l$. A direct tally on the five ITC days and the Lancaster dataset used here confirms this: between 44% and 67% of incompatible (lecture, hall) pairs on the ITC instances involve a hall whose capacity already satisfies $u_h \geq s_l$ and that is nonetheless excluded from $H(l)$, and on Lancaster *every* hall excluded from some $H(l)$ has sufficient capacity, so capacity is never the binding driver of incompatibility there. Equivalently, the average fraction of halls with $u_h \geq s_l$ that are excluded from $H(l)$ for other reasons ranges from 65% to 95% on ITC and reaches 97% on Lancaster. Because the capacity-dominance cut exploits a stratification of halls by capacity relative to a threshold κ , it has little to bite on inputs in which $H(l)$ has already been pruned by factors largely uncorrelated with capacity, which is consistent with the small deltas observed in Table 6. One would therefore expect the cut to be substantially more effective on inputs where hall capacity is the dominant compatibility driver, for example institutions with a near-homogeneous feature profile or instances that apply only the generic filter $H(l) = \{h : u_h \geq s_l\}$.

Taken together, the results suggest a clear hierarchy among the three improvements: biclique strengthening is the dominant source of bound improvement and exact-solver speed, compatibility preprocessing is a consistently mild positive, and the Chvátal–Gomory capacity-dominance cuts provide no measurable benefit on the 35 real-world instances studied here and should be disabled by default on this benchmark.

5. Conclusions and future research

This paper studied the *lecture-hall assignment problem* that arises once a weekly timetable is fixed and student enrollments are known. The problem asks to assign each scheduled lecture to a feasible hall, respecting capacity and interval-overlap constraints, so as to minimize the total walking effort induced by students moving between consecutive lectures. This quadratic objective, driven by the product of shared enrollments and hall-to-hall distances, is absent from most of the timetabling and hall-allocation literature and yet captures a first-order operational concern at real universities, where long between-class walks generate late arrivals and place a disproportionate burden on students with mobility needs.

Our main contributions are fourfold. First, on the modeling side, we formally isolated the post-timetabling hall assignment layer from the broader university course timetabling problem and proved that, even in a restricted single-day two-block setting with identical capacities and full compatibility, the decision version is NP-complete and every approximation lower bound for the quadratic assignment problem carries over. This places the problem squarely in the QAP complexity landscape while clarifying that the usual interval-scheduling relaxations are insufficient to capture its difficulty. Second, we developed three exact formulations, a bilinear MIQP, a compact linearized MIP with one continuous distance variable per successor pair, and a CP-SAT formulation based on integer hall-assignment variables, all of which share the same feasible region and decompose by day. Third, we introduced three problem-specific modeling improvements that can be layered on top of any of the three formulations: an easy-to-separate family of *biclique distance cuts* that dominates the basic pairwise linking inequalities and exploits the sparsity of lecture-hall compatibilities, a family of Chvátal–Gomory *capacity-dominance* inequalities that combine clique-overlap constraints with hall-capacity thresholds, and a safe fixed-point *compatibility-preprocessing* procedure. We also showed that the framework extends to standard timetabling side constraints such as hard and soft *Same-Room* and *SameAttendees*, for which the biclique construction again delivers stronger aggregated inequalities. Fourth, we conducted an extensive computational study on 35 daily instances derived from five large ITC 2019 benchmarks and from a 2023 institutional dataset at Lancaster University, totaling 840 exact runs under a 300-second time limit.

The numerical results point to a clear hierarchy among the proposed enhancements. The compact linearized MIP is the most robust exact formulation on the benchmark, solving 23 of the 25 ITC daily instances and all 10 Lancaster instances to proven optimality within the time limit, and achieving an average best-known gap of 0.09% on ITC and 0.00% on Lancaster in the representative block. Biclique strengthening is the dominant single modeling device: it reduces the mean root-relaxation shortfall of the linearized model from 5.76% to 0.09% on ITC instances and from 20.67% to 0.01% on Lancaster instances, and lifts MIP optimal solves from 16/25 to 23/25 on ITC while turning the weak Lancaster MIP, which times out on every instance, into a model that solves all ten in under a second of median runtime. In contrast, light compatibility preprocessing contributes a consistent modest speed-up across all configurations, while the Chvátal–Gomory capacity-dominance cuts, although theoretically sound, provide no measurable benefit on the 35 real-world instances studied here: they are essentially neutral for MIQP and for every Lancaster configuration, and net negative for the MIP and CP-SAT formulations on ITC. The CP-SAT formulation produces strong incumbents but is consistently slower at proving optimality, and the MIQP formulation is competitive on easier instances but loses ground on the hardest Purdue days, where the nonlinear branch-and-bound under stronger cuts sometimes fails to return any incumbent within the time limit.

Taken together, these findings suggest that realistic lecture-hall assignment problems from both a competition-grade benchmark and an institutional dataset can be solved to optimality or near-optimality within minutes by off-the-shelf MIP technology once the biclique structure of the quadratic

walking term is exploited. From a practical standpoint, this places the proposed framework within reach of operations offices that are already comfortable with mathematical-programming software, and it complements the multi-objective evolutionary approaches that target the closely related but broader hall-allocation problem (Antunes-Batista et al., 2026).

Several directions for future research appear promising. On the algorithmic side, the present biclique inequalities are separated in a static, structural manner; it would be natural to study their polyhedral strength more formally, identify classes under which they are facet-defining, and develop a dynamic separation routine embedded in a branch-and-cut scheme. Complementing this, dedicated Lagrangian or convex quadratic relaxations of the walking term, along the lines of Anstreicher (2003) and Zhao et al. (1998), may yield even tighter bounds on the hardest instances, and combining the linearized MIP with tabu-style heuristics inspired by Taillard (1991) could provide high-quality warm starts for very large instances. A second direction is to relax the assumption of a fully fixed timetable and allow limited schedule adjustments, such as swapping two lectures of the same course between adjacent time slots, which would couple the hall-assignment decision with light timetable repair while preserving much of the structural decomposition by day. Third, a rich class of robust and stochastic extensions becomes interesting once one recognizes that enrollments, disability-related weights, and hall availability are often uncertain at the time of assignment; two-stage stochastic formulations that choose a hall assignment before enrollment realization, together with appropriate scenario-based bounds, would be a natural next step. Fourth, the framework could be embedded in a digital-twin or decision-support system for campus mobility, jointly accounting for elevator and stair bottlenecks, weather and season effects on outdoor walking, and accessibility constraints for students with disabilities; the linear assignment-penalty extension already accommodates such weightings. Finally, the multi-criteria structure explored by Antunes-Batista et al. (2026) and the hybrid-teaching dimension emphasized by Davison et al. (2025) remain attractive directions: combining their broader objective landscape with the exact-optimization machinery proposed here may yield decision-support tools that are both provably optimal with respect to a primary walking objective and explicitly Pareto-aware across secondary criteria.

Declaration of competing interests

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data and code availability

This preprint is accompanied by a public replication package. The complete code base, its documentation, and the detailed result files for all 1120 Gurobi and CP-SAT runs underlying the numerical campaign are publicly available at <https://github.com/ta169/lecture-halls>. The ITC 2019 source XML files are publicly distributed through the ITC 2019 repository of Müller et al. (2025). The Lancaster 2023 source data were kindly provided by Matthew Davison for research use and are available from him upon reasonable request, subject to the institution’s data-sharing conditions.

Acknowledgements

The author is grateful to Matthew Davison for sharing the Lancaster 2023 timetabling dataset that underpins a substantial portion of the computational study.

References

- Abdipoor, S., Yaakob, R., Goh, S.L., Abdullah, S., 2023. Meta-heuristic approaches for the university course timetabling problem. *Intelligent Systems with Applications* 19, 200253. doi:10.1016/j.iswa.2023.200253.
- Abdullah, S., Turabieh, H., 2012. On the use of multi-neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences* 191, 146–168. doi:10.1016/j.ins.2011.12.018.
- Adams, W.P., Johnson, T.A., 1994. Improved linear programming-based lower bounds for the quadratic assignment problem, in: *Quadratic Assignment and Related Problems*. American Mathematical Society. volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 43–75.
- Anstreicher, K., Brixius, N., Goux, J.P., Linderoth, J., 2002. Solving large quadratic assignment problems on computational grids. *Mathematical Programming* 91, 563–588. doi:10.1007/s101070100260.
- Anstreicher, K.M., 2003. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming* 97, 27–42. doi:10.1007/s10107-003-0437-z.
- Antunes-Batista, B., Atta, S., Basto-Fernandes, V., Yevseyeva, I., Emmerich, M.T.M., 2026. Multi-objective optimization approaches for room allocation in university course timetabling. *Procedia Computer Science* 278, 663–670. doi:10.1016/j.procs.2026.03.037.
- Bonutti, A., De Cesco, F., Di Gaspero, L., Schaerf, A., 2012. Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research* 194, 59–70. doi:10.1007/s10479-010-0707-0.
- Burer, S., Vandenbussche, D., 2006. Solving lift-and-project relaxations of binary integer programs. *SIAM Journal on Optimization* 16, 726–750. doi:10.1137/040609574.
- Burkard, R.E., Çela, E., Pardalos, P.M., Pitsoulis, L.S., 1998. The quadratic assignment problem, in: *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, pp. 241–338.
- Burke, E.K., Petrovic, S., 2002. Recent research directions in automated timetabling. *European Journal of Operational Research* 140, 266–280. doi:10.1016/S0377-2217(02)00069-3.
- Carter, M.W., Tovey, C.A., 1992. When is the classroom assignment problem hard? *Operations Research* 40, S28–S39. doi:10.1287/opre.40.1.S28.
- Davison, M., Kheiri, A., Zografos, K.G., 2025. Modelling and solving the university course timetabling problem with hybrid teaching considerations. *Journal of Scheduling* 28, 195–215. doi:10.1007/s10951-024-00817-w.
- Er-rhaimini, K., 2020. Forest growth optimization for solving timetabling problems. Solver summary paper on the official ITC 2019 website. URL: <https://www.itc2019.org/papers/itc2019-er-rhaimini.pdf>. accessed April 17, 2026.

- Fischer, A., Fischer, F., Hungerländer, P., 2019. New exact approaches to row layout problems. *Mathematical Programming Computation* 11, 703–754. doi:10.1007/s12532-019-00162-6.
- Frieze, A.M., Yadegar, J., 1983. On the quadratic assignment problem. *Discrete Applied Mathematics* 5, 89–98. doi:10.1016/0166-218X(83)90018-5.
- Gilmore, P.C., 1962. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics* 10, 305–313. doi:10.1137/0110022.
- Hahn, P., Grant, T., 1998. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research* 46, 912–922. doi:10.1287/opre.46.6.912.
- Holm, D.S., Mikkelsen, R.Ø., Sørensen, M., Stidsen, T.R., 2020. A mip based approach for international timetabling competition 2019. Solver summary paper on the official ITC 2019 website. URL: <https://www.itc2019.org/papers/itc2019-holm.pdf>. accessed April 17, 2026.
- Kaufman, L., Broeckx, F., 1978. An algorithm for the quadratic assignment problem using Bender’s decomposition. *European Journal of Operational Research* 2, 207–211. doi:10.1016/0377-2217(78)90095-4.
- Kolen, A.W.J., Lenstra, J.K., Papadimitriou, C.H., Spiessma, F.C.R., 2007. Interval scheduling: A survey. *Naval Research Logistics* 54, 530–543. doi:10.1002/nav.20231.
- Koopmans, T.C., Beckmann, M., 1957. Assignment problems and the location of economic activities. *Econometrica* 25, 53–76. doi:10.2307/1907742.
- Lawler, E.L., 1963. The quadratic assignment problem. *Management Science* 9, 586–599. doi:10.1287/mnsc.9.4.586.
- Lewis, R., Paechter, B., 2005. Application of the grouping genetic algorithm to university course timetabling, in: *Evolutionary Computation in Combinatorial Optimization (EvoCOP 2005)*, Springer. pp. 144–153. doi:10.1007/978-3-540-31996-2_14.
- Loiola, E.M., de Abreu, N.M.M., Boaventura-Netto, P.O., Hahn, P., Querido, T., 2007. A survey for the quadratic assignment problem. *European Journal of Operational Research* 176, 657–690. doi:10.1016/j.ejor.2005.09.032.
- McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Di Gaspero, L., Qu, R., Burke, E.K., 2010. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing* 22, 120–130. doi:10.1287/ijoc.1090.0320.
- Müller, T., Rudová, H., Müllerová, Z., 2020. International timetabling competition 2019 awards slides. Official ITC 2019 website. URL: <https://www.itc2019.org/papers/itc2019-awards-slides.pdf>. accessed April 17, 2026.
- Müller, T., Rudová, H., Müllerová, Z., 2025. Real-world university course timetabling at the international timetabling competition 2019. *Journal of Scheduling* 28, 247–267. doi:10.1007/s10951-023-00801-w.
- Rappos, E., Thi’emard, E., Robert, S., Hèche, J.F., 2022. A mixed-integer programming approach for solving university course timetabling problems. *Journal of Scheduling* 25, 391–404. doi:10.1007/s10951-021-00715-5.

- Sahni, S., Gonzalez, T., 1976. P-complete approximation problems. *Journal of the ACM (JACM)* 23, 555–565.
- Schaerf, A., 1999. A survey of automated timetabling. *Artificial Intelligence Review* 13, 87–127. doi:10.1023/A:1006576209967.
- Taillard, E., 1991. Robust taboo search for the quadratic assignment problem. *Parallel Computing* 17, 443–455. doi:10.1016/S0167-8191(05)80147-4.
- Torres-Ovalle, C., Montoya-Torres, J.R., Quintero-Araujo, C., Sarmiento, A., 2014. University course scheduling and classroom assignment. *Ingeniería y Universidad* 18, 59–75. doi:10.11144/Javeriana.IYU18-1.ucsc.
- Zhao, Q., Karisch, S.E., Rendl, F., Wolkowicz, H., 1998. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization* 2, 71–109. doi:10.1023/A:1009795911987.

Appendix A. Detailed per-instance numerical results

This appendix reports, for each of the 35 real-world benchmark instances used in Section 4, the per-instance counterparts of the aggregate numbers in Tables 3 and 4. Tables A.7 and A.8 give the exact-solve results for the three formulations on the representative configuration (light compatibility preprocessing, no capacity-dominance cuts, biclique strengthening enabled except for the ITC-MIQP column, which uses the weaker pairwise distance model for the reasons discussed in Section 4). Tables A.9 and A.10 give the corresponding root-relaxation results for the weak pairwise and the strong biclique-enhanced distance models. Each exact row reports, per formulation, the final incumbent objective value, the solver-certified optimality gap (%), the best-known gap (%), and the solver runtime (s). The best-known gap is computed against the global LB^* column, $bk\% = 100(z - LB^*)/LB^*$, and it coincides with the certified gap whenever the run attained LB^* on its own, as is the case on every ITC `agh`, `muni-pdf`, `muni-pdfx` instance and on every Lancaster instance in the representative block. A dagger ([†]) marks runs that hit the 300-second time limit: the solver stopped with a feasible but suboptimal incumbent and a valid lower bound that was not tight enough to close the gap inside the time budget. In every run reported in this appendix the solver returned a positive lower bound, so the gap columns are always well-defined. The column LB^* is the best lower bound obtained by any method for that instance, and serves as the reference point for the shortfall columns in the root tables. The per-instance patterns mirror the aggregate summary: the linearized MIP closes the gap on essentially every instance, the quadratic MIP also stays close to LB^* once the ITC-MIQP exception is taken into account, and CP-SAT closes all Lancaster instances but struggles on the large Purdue instances. On the root relaxations, enabling biclique distance strengthening reduces the shortfall to within a fraction of a percent on almost every instance and turns a solver that times out on many instances into one that returns a certified root bound in a few tenths of a second.

A comprehensive report on the results of all the 1120 Gurobi and CP-SAT runs that comprise our numerical campaign can be downloaded from <https://github.com/tal69/lecture-halls>.

Table A.7: Per-instance exact-solve results on the 25 ITC 2019 benchmark days. All rows use light compatibility preprocessing and no capacity-dominance cuts; the MIP and CP columns use biclique distance strengthening, whereas the MIQP column uses the weaker pairwise distance model for the reasons discussed in Section 4. Sub-column “gap%” is the solver-certified gap against that run’s own lower bound, and “bk%” is the best-known gap $(z - LB^*)/LB^*$ against the global reference in the third column. Instance names abbreviate **agh-fal17**, **muni-pdf-spr16c**, **muni-pdfx-fal17**, **pu-d9-fal19**, and **pu-proj-fal19**; week and day indices are encoded as **wXdY**. † time limit reached: the solver returned a feasible but suboptimal incumbent with a valid lower bound that was not tight enough to certify optimality inside the 300-second budget.

Instance	L	LB*	MIQP				MIP				CP			
			obj	gap%	bk%	t (s)	obj	gap%	bk%	t (s)	obj	gap%	bk%	t (s)
agh w4d1	542	966	966	0.00	0.00	0.39	966	0.00	0.00	0.06	966	0.00	0.00	6.8
agh w4d2	558	649	649	0.00	0.00	0.37	649	0.00	0.00	0.06	649	0.00	0.00	9.1
agh w4d3	540	509	509	0.00	0.00	0.24	509	0.00	0.00	0.08	509	0.00	0.00	5.3
agh w4d4	537	523	523	0.00	0.00	0.39	523	0.00	0.00	0.07	523	0.00	0.00	6.3
agh w4d5	540	559	559	0.00	0.00	0.22	559	0.00	0.00	0.08	559	0.00	0.00	4.2
muni-pdf w3d1	217	85	85	0.00	0.00	0.00	85	0.00	0.00	0.00	85	0.00	0.00	0.00
muni-pdf w3d2	222	143	143	0.00	0.00	0.00	143	0.00	0.00	0.00	143	0.00	0.00	0.00
muni-pdf w3d3	217	148	148	0.00	0.00	0.00	148	0.00	0.00	0.00	148	0.00	0.00	0.00
muni-pdf w3d4	192	120	120	0.00	0.00	0.00	120	0.00	0.00	0.00	120	0.00	0.00	0.00
muni-pdf w3d5	159	306	306	0.00	0.00	0.32	306	0.00	0.00	0.03	306	0.00	0.00	2.2
muni-pdfx w12d1	361	1611	1611	0.00	0.00	0.86	1611	0.00	0.00	0.05	1611	0.00	0.00	39
muni-pdfx w12d2	356	1543	1543	0.00	0.00	0.71	1543	0.00	0.00	0.13	1543	0.00	0.00	45
muni-pdfx w12d3	388	2365	2365	0.00	0.00	1.0	2365	0.00	0.00	0.06	2365	0.00	0.00	57
muni-pdfx w12d4	332	1469	1469	0.00	0.00	0.59	1469	0.00	0.00	0.05	1469	0.00	0.00	87
muni-pdfx w12d5	197	909	909	0.00	0.00	0.53	909	0.00	0.00	0.05	909	0.00	0.00	2.6
pu-d9 w2d1	909	12301	12304	5.70	0.02	300†	12302	0.00	0.01	37	12719	50.50	3.40	301†
pu-d9 w2d2	846	6066	6066	0.00	0.00	2.7	6066	0.00	0.00	0.61	6066	0.00	0.00	65
pu-d9 w2d3	977	12654	12655	7.15	0.01	300†	12655	0.00	0.01	263	13042	40.95	3.07	301†
pu-d9 w2d4	857	4902	4902	0.00	0.00	15	4902	0.00	0.00	8.1	4907	2.55	0.10	300†
pu-d9 w2d5	883	10944	10945	1.97	0.01	300†	10945	0.00	0.01	23	11125	34.08	1.65	301†
pu-proj w6d1	2348	24076	24479	16.97	1.67	300†	24317	1.10	1.00	300†	30195	94.82	25.42	302†
pu-proj w6d2	2362	10078	10078	0.39	0.00	300†	10078	0.00	0.00	10	10099	24.82	0.21	300†
pu-proj w6d3	2562	24382	24781	21.45	1.64	300†	24652	1.11	1.11	300†	34347	126.94	40.87	302†
pu-proj w6d4	2320	9980	9980	0.10	0.00	300†	9980	0.00	0.00	3.9	10101	16.68	1.21	300†
pu-proj w6d5	2208	21373	21385	8.92	0.06	300†	21373	0.00	0.00	221	24920	61.19	16.60	302†

Table A.8: Per-instance exact-solve results on the 10 Lancaster 2023 benchmark days. All rows use the representative configuration with biclique distance strengthening enabled for all three formulations. Sub-column “bk%” is the best-known gap $(z - LB^*)/LB^*$ and equals the certified gap on every row because every formulation reaches LB^* . Instance names abbreviate **lancs-yr23_term1** and **lancs-yr23_term2**; week and day indices are encoded as **wXdY**.

Instance	L	LB*	MIQP				MIP				CP			
			obj	gap%	bk%	t (s)	obj	gap%	bk%	t (s)	obj	gap%	bk%	t (s)
lancs t1 w13d1	528	4319	4319	0.00	0.00	22	4319	0.00	0.00	0.37	4319	0.00	0.00	57
lancs t1 w13d2	528	2585	2585	0.00	0.00	14	2585	0.00	0.00	0.27	2585	0.00	0.00	43
lancs t1 w13d3	306	1650	1650	0.00	0.00	8.5	1650	0.00	0.00	0.14	1650	0.00	0.00	10
lancs t1 w13d4	587	2848	2848	0.00	0.00	31	2848	0.00	0.00	1.4	2848	0.00	0.00	49
lancs t1 w13d5	510	2866	2866	0.00	0.00	26	2866	0.00	0.00	0.61	2866	0.00	0.00	35
lancs t2 w28d1	548	3105	3105	0.00	0.00	43	3105	0.00	0.00	1.9	3105	0.00	0.00	62
lancs t2 w28d2	533	3944	3944	0.00	0.00	61	3944	0.00	0.00	6.8	3944	0.00	0.00	41
lancs t2 w28d3	312	1860	1860	0.00	0.00	5.5	1860	0.00	0.00	0.16	1860	0.00	0.00	13
lancs t2 w28d4	597	2673	2673	0.00	0.00	30	2673	0.00	0.00	1.8	2673	0.00	0.00	57
lancs t2 w28d5	546	2772	2772	0.00	0.00	48	2772	0.00	0.00	0.58	2772	0.00	0.00	68

Table A.9: Per-instance root-relaxation results for the linearized MIP on the 25 ITC 2019 benchmark days. “Weak” denotes the LP relaxation of the pairwise distance model used as baseline in the root experiments; “Strong” denotes the biclique-enhanced distance model. Shortfall is measured as $(LB^* - LB)/LB^*$. Reported runtimes include root cut generation; [†] time limit reached.

Instance	LB^*	Weak (pairwise)			Strong (biclique)		
		LB	shortfall%	t (s)	LB	shortfall%	t (s)
agh w4d1	966	965	0.10	0.46	966	0.00	0.07
agh w4d2	649	644	0.77	0.43	649	0.00	0.06
agh w4d3	509	509	0.00	0.24	509	0.00	0.09
agh w4d4	523	519	0.76	0.74	521	0.38	0.08
agh w4d5	559	559	0.00	0.30	559	0.00	0.08
muni-pdf w3d1	85	85	0.00	0.00	85	0.00	0.00
muni-pdf w3d2	143	143	0.00	0.00	143	0.00	0.00
muni-pdf w3d3	148	148	0.00	0.00	148	0.00	0.00
muni-pdf w3d4	120	120	0.00	0.00	120	0.00	0.00
muni-pdf w3d5	306	306	0.00	0.41	306	0.00	0.03
muni-pdfx w12d1	1611	1611	0.00	1.5	1611	0.00	0.05
muni-pdfx w12d2	1543	1523	1.30	2.8	1543	0.00	0.13
muni-pdfx w12d3	2365	2365	0.00	1.9	2365	0.00	0.06
muni-pdfx w12d4	1469	1469	0.00	2.4	1469	0.00	0.06
muni-pdfx w12d5	909	909	0.00	0.88	909	0.00	0.06
pu-d9 w2d1	12301	10659	13.35	300 [†]	12301	0.00	37
pu-d9 w2d2	6066	5951	1.90	14	6066	0.00	0.61
pu-d9 w2d3	12654	10712	15.35	300 [†]	12602	0.41	58
pu-d9 w2d4	4902	4671	4.71	28	4872	0.61	4.2
pu-d9 w2d5	10944	9346	14.60	224	10944	0.00	23
pu-proj w6d1	24076	17434	27.59	300 [†]	24031	0.19	153
pu-proj w6d2	10078	9269	8.03	125	10065	0.13	10
pu-proj w6d3	24382	17724	27.31	300 [†]	24369	0.05	147
pu-proj w6d4	9980	9069	9.13	144	9980	0.00	3.9
pu-proj w6d5	21373	17282	19.14	300 [†]	21295	0.36	48

Table A.10: Per-instance root-relaxation results for the linearized MIP on the 10 Lancaster 2022/23 benchmark days. Notation as in Table A.9.

Instance	LB^*	Weak (pairwise)			Strong (biclique)		
		LB	shortfall%	t (s)	LB	shortfall%	t (s)
lancs t1 w13d1	4319	3460	19.89	54	4319	0.00	0.38
lancs t1 w13d2	2585	2221	14.08	46	2585	0.00	0.28
lancs t1 w13d3	1650	1485	10.00	22	1650	0.00	0.15
lancs t1 w13d4	2848	2027	28.83	59	2848	0.00	1.4
lancs t1 w13d5	2866	2363	17.55	40	2866	0.00	0.64
lancs t2 w28d1	3105	2486	19.94	91	3102	0.10	2.0
lancs t2 w28d2	3944	2963	24.87	64	3944	0.00	6.9
lancs t2 w28d3	1860	1550	16.67	21	1860	0.00	0.14
lancs t2 w28d4	2673	2047	23.42	57	2673	0.00	1.9
lancs t2 w28d5	2772	1901	31.42	78	2772	0.00	0.60