



HAL
open science

Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems

Taha Arbaoui, Jean-Paul Boufflet, Aziz Moukrim

► To cite this version:

Taha Arbaoui, Jean-Paul Boufflet, Aziz Moukrim. Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems. *Computers and Operations Research*, 2019, 106, pp.133-142. 10.1016/j.cor.2019.02.013 . hal-02080090

HAL Id: hal-02080090

<https://hal.science/hal-02080090>

Submitted on 24 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Lower bounds on spacing soft constraints for university examination timetabling problems, and compact mathematical formulations

Taha Arbaoui ^{*1}, Jean-Paul Boufflet ^{† 2}, and Aziz Moukrim ^{‡2}

¹Laboratoire d'Optimisation des Systèmes Industriels, Institut Charles Delaunay (ICD-LOSI), UMR CNRS 6281, Université de Technologie de Troyes, 12 rue Marie Curie, CS 42060, 10004, Troyes, France

²Sorbonne universités, Université de Technologie de Compiègne, Laboratoire Heudiasyc, UMR CNRS 7253, 57 avenue de Landshut, CS 60319, 60203 Compiègne, France

Abstract

The examination timetabling problem (ETP) can be described as a set of exams to be scheduled over an examination session while respecting numerous hard and soft constraints. Spacing soft constraints, which seek to spread the exams taken by students out over the available periods rather than scheduling them close together, are widely encountered in real-world ETP problems. In this paper we consider the spacing soft constraints that seek to prevent students sitting more than one exam a day, for reason of fairness, as defined in the ITC2007 examination timetabling track. Work on ETP has tended to focus on heuristic approaches, and little effort has gone developing lower bounds, although knowing how far a given solution is from optimality is of both practical and theoretical interest. In existing mathematical formulations of spacing soft constraints the number of equations is in the order of the square of the number of exams. These formulations consume excessive quantities of memory, which remains a problem for current solvers. In this study, we present a generic model for computing lower bounds on spacing soft constraints, together with more compact formulations where the number of equations is equal to the number of exams, and not in the order of the square of this number. Computational results on ITC2007 and Yeditepe instances are an improvement on results obtained so far on lower bounds, and our new formulations yield a more compact model achieving better results compared to existing formulations. Some effort still needs be devoted to lower bounds and to exact methods capable of bridging the gap between theory and practice.

Keywords: Examination timetabling problems, Spacing soft constraints, Lower bounds, Compact mathematical formulation

*taha.arbaoui@utt.fr

†jean-paul.boufflet@hds.utc.fr

‡aziz.moukrim@hds.utc.fr

1 Introduction

Academic institutions periodically face the problem of organizing examination sessions. Designing an examination timetable is a complex, time consuming and tedious task for practitioners that quite often have to deal with contradictory expectations of the institution, teachers, and students. The examination timetabling problem (ETP) can be described as a set of exams to be scheduled into a set of periods while respecting numerous hard and soft constraints. The quality of a solution that satisfies all the hard constraints is measured by assessing violations of soft constraints. A large number of variants of ETPs can be found in the literature, every institution has its own set of hard constraints and uses a different set of soft constraints to assess solutions.

In 1997, Burke et al. [10] presented an introduction to automated examination timetabling using information collected within UK Universities, cataloging and structuring common types of constraints, and popular approaches used at the time. In a latter comprehensive survey, Schaerf [30] presented school, course and examination timetabling problems. Here, variants of ETPs were discussed and solutions were reviewed and compared. Many methodologies were developed to tackle specific problems, however scientific comparisons are difficult because presented problems do not have the same sets of hard and soft constraints. Benchmarks with standard variants from the Universities of Toronto [11], Nottingham [15] and Melbourne [23] were proposed that make comparisons possible. In 2009, Qu et al. [28] summarized earlier surveys on examination timetabling [10, 12, 14, 30] and presented the state of the art on solution methods. Benchmarks available at that time were presented, approaches and results on the benchmark datasets were discussed. The second International Timetabling Competition (ITC2007) examination timetabling track [21] introduced a problem description which covers many real-world situations. These benchmarks are at the research community's disposal to permit comparisons.

Examination timetabling has been intensively studied over the last decades using a large variety of optimisation techniques for preprocessing and for solving.

Gogos et al. [19] and Arbaoui et al. [7] proposed preprocessing to reveal hidden hard constraints that can be deduced prior to solving. The initial data can then be enriched and infeasibilities can be detected sooner during the solving process.

A large number of solution methods based on heuristics, meta-heuristics and their hybridizations, and hyper-heuristics can be found in the literature. A selection of investigated methods includes: Graph Ordering Heuristic [2], Tabu Search [20], Simulated Annealing [16], Great Deluge [25], Hill Climbing with or without Late Acceptance [8], Bin Packing Heuristic [31], Evolutionary and Nature Inspired Algorithms [3] and Hyper-Heuristic [9]. Recently, Alzaqebah et al. [5] obtained the best result found so far for one instance of the Toronto benchmark using a hybrid bee colony approach, simulated annealing and late acceptance hill climbing.

Exact methods or their hybridization with heuristics have been studied to solve examination timetabling problems. Hybridizing integer programming and

a decomposition approach was investigated by Qu et al. [29]. An integer programming phase for assigning exams to rooms was used by Gogos et al. [19]. For a specific problem, MirHassani [24] proposed a Mixed Integer Programming (MIP) model. A three-stage Integer Linear Programming (ILP) approach is proposed by Al-Hawari et al. [4] to solve the examination timetabling problem at the German Jordanian University. Recently, a column generation approach was proposed by Woumans et al. [32]. To increase the spacing of exams for students, multiple versions of an exam to be planned are allowed, and a trade-off between spacing for students and numbers of exam versions is optimized.

In this paper we consider the ITC2007 problem formulation. The objective function for the ITC2007 benchmark is a sum of seven terms that can interact in contradictory ways. One of the terms assesses allocations of large exams to periods at the end of the examination session because large exams take longer to mark (teacher expectations). Large exams may not be spread out within all periods, this may act in a contradictory way with other terms that assess spacing soft constraint violations (student expectations). The objective function is a measure of a trade-off between expectations of the institution, teachers, and students.

The results have been steadily improved over the last decade by applying heuristic approaches. Solutions are obtained but we do not know how far from optimality these solutions actually are.

Real-world problems may contain subsets of ITC2007 soft constraints. The results presented in the literature provide upper bounds for the value of the objective function but we do not have lower bounds for each term considered individually. It could also be practically and theoretically interesting to assess lower bounds on terms that aim to reach the same objective. The spacing soft constraints are widely encountered in real-world ETP problems. In particular, it is preferable for students not to have to sit more than one exam a day.

To the best of our knowledge, Arbaoui et al. [6] presented the first work on spacing soft constraint lower bounds for ITC2007 problems. A set of selected cliques, computed on the conflict graph coded using the conflict matrix (see Carter et al. [15]), is used to compute lower bounds.

Few exact methods have been reported compared to heuristic based approaches. McCollum et al. [22] proposed a new mathematical model for the ITC2007 examination timetabling track that permits to have a meaningful basis for hard and soft constraints. As stated by the authors, the model was not designed to solve sizeable real instances but only for modeling purposes. However, it can optimally solve two small instances of the Yeditepe datasets [27]. Fonseca et al. [17] proposed an improved version of this model that fits better in memory and less out-of-memory were observed on sizeable instances. An improved formulation proposed by Arbaoui et al. [7] was run on each term considered individually. For certain terms optimal values were attained. However, current solvers faced difficulties to run for assessing spacing soft constraints terms.

Soghier et al. [31] investigated bin packing heuristics and proposed an adap-

tive hybrid hyper-heuristic approach. This way of approaching ETP considers exams are items and rooms/periods are bins. Data-dependent Dual-Feasible Functions (DDFF) were proposed for building lower bounds for bin-packing problems (see Carlier et al. [13]). In [7], the authors proposed a valid inequality based on an adaption of a DDFF that proved to be efficient on small instances.

Applying valid inequalities while reducing the number of variables and constraints proved to be helpful improving exact methods based on integer programming formulations [7, 17]. However, the spacing soft constraint formulations consume excessive quantities of memory and current solvers still face difficulties to run models.

In this study, we present improvements for computing two spacing soft constraint lower bounds, together with more compact formulations yielding to design a MIP model that consume less memory.

The rest of the paper is organized as follows. In the next section, hard and soft constraints of the ITC2007 examination timetabling competition are presented. In Section 3, we propose three propositions to compute minimum number of spacing soft constraint violations for every clique of a given size, and the generic MIP model that computes the associated lower bounds. In Section 4, the new formulations of the three spacing soft constraints are described, the model embedding these formulations is compared to existing MIP models. In Section 5, computational results assessed on ITC2007 [21] and Yeditepe [27] datasets are presented and commented on, and concluding remarks are given in Section 6.

2 Problem formulation

This section provides a description for the examination track of the Second International Timetabling Competition (ITC2007) [21] used in our study.

Input data consists of exams, rooms and days/periods. Every exam has a duration, and has to be allocated to a period and a room. The set of students enrolled for each exam is given. Each room has a seating capacity, and some rooms can have a penalty for their use. Exams can share a room, as long as the capacity of the room is respected. The examination session is composed of days defined over a specified length of time. Each day has a number of periods, each period has a duration, and some periods can have a penalty for their use.

The hard constraints are:

Conflicts: A student can sit only one exam at a time.

Period length: The exam duration must be less than or equal to the period duration.

Room capacity. The capacity of any room cannot be exceeded at any period.

Room exclusivity: An exam must take place in a room on its own.

Time-ordering: An order between pair of exams must be respected.

The time-ordering hard constraints are **After** constraint, an exam i has to be scheduled after exam j , **Exclusion** constraint, exams i and j have to be scheduled at different periods, and **Coincidence** constraint, exams i and j must be scheduled to the same period.

The soft constraints used to assess the terms of the objective function are:

Front-Load: Exams having a large number of students should be allocated in the earlier periods of the session.

Period-Penalty: An exam should not be allocated to a penalized period.

Room-Penalty: An exam should not be allocated to a penalized room.

Non-Mixed-Duration: Exams allocated in the same period to the same room should have the same duration.

Two-In-a-Row: For a student, exams that are scheduled back-to-back in the same day should be avoided.

Two-In-a-Day: For a student, exams that are scheduled in the same day but not back-to-back should be avoided.

Period-Spread: For a student, exams should be spaced out over a fixed number of periods.

Data and weights that involving assessing the soft constraints are provided in the input file of each instance.

Two-In-a-Row, *Two-In-a-Day* and *Period-Spread* are spacing soft constraints that aim at spreading out exams taken by students over the exam session.

3 Two-In-a-Row & Two-In-a-Day lower bounds

Here, we propose improvements for calculating lower bounds for *Two-In-a-Row* and *Two-In-a-Day* spacing soft constraints. We consider them both individually and together since they aim to prevent students sitting more than one exam a day.

To the best of our knowledge, the first attempt to design lower bounds for these soft constraints was proposed by Arbaoui et al. [6]. The idea is to use cliques calculated on the conflict graph coded using the conflict matrix (see Carter et al. [15]). The authors proposed three limits on the size of a clique beyond which at least one violation of these spacing constraints occurs. Cliques of sizes larger than these limits are selected. For every clique, contributions for penalties of these spacing soft constraints are computed using a MIP formulation that permits to obtain optimal values. The contributions of two cliques with no common edge can be summed. To compute a lower bound, a family of edge-disjoint cliques is built applying a greedy heuristic.

A greedy heuristic building a family of edge-disjoint cliques provides an evaluation without any guarantee of optimality. Moreover, since selected cliques have no common edges, some edges cannot be counted as violations. Furthermore, the greedy heuristic does not take account of *Coincidence* constraints between exams.

We propose calculation of minimum numbers of violations in Section 3.1. The limits allow us to know whether a clique induces at least one soft constraint violation, but they do not allow minimum numbers of violations to be computed.

In Section 3.2, we propose a generic MIP model to compute lower bounds for *Two-In-a-Row* and *Two-In-a-Day* soft constraints (individually or together). The formulation we propose allows lower bounds to be computed optimally using minimum numbers of violations and individual contributions of selected cliques. This formulation ensures that a violation cannot be counted twice whilst taking into account *Coincidence* constraints between exams.

3.1 Minimum number of violations

For a clique c of size k of the conflict graph, we propose to compute $\lambda^R(k)$, $\lambda^D(k)$ and $\lambda^{RD}(k)$, the minimum numbers of *Two-In-a-Row*, *Two-In-a-Day*, or *Two-In-a-Row* and *Two-In-a-Day* spacing soft constraint violations.

The conflict graph $G(E, A_C)$ is defined as follows: E is the set of exams, and for exams i and j , there is an edge $[i, j] \in A_C$ if at least a student takes the two exams. Each edge $[i, j]$ is weighted by w_{ij}^C , the number of students taking exams i and j . Every spacing soft constraint violation is assessed using the weight of the associated edge. The number of soft constraint violations equals the number of weighted edges to be summed to assess either the *Two-In-a-Row* term or the *Two-In-a-Day* term of the objective function.

A day is said to be of type Di if it has i periods, and n^{Di} is the number of days of type Di . Not all the possible Di 's exist for a particular instance. We denote δ as the set of numbers of periods that corresponds to the types of days of an instance. As an example, $\delta = \{2, 3, 4\}$ means that we have types of days $D2$, $D3$ and $D4$. Then, we denote $\varphi = \sum_{d \in (\delta \setminus \{1\})} n^{Dd}$ the sum of numbers of $D2$, $D3$ and $D4$ days. Note that there are no $D5$ days, neither in the ITC2007 nor in the Yeditepe instances.

A clique corresponds to pairwise adjacent exams in the conflict graph, so these exams must be allocated in different periods. The three limits on the size of a clique beyond which at least one violation of *Two-In-a-Row*, or *Two-In-a-Day*, or *Two-In-a-Row* and *Two-In-a-Day* occurs are the following:

$$\begin{aligned} L^R &= \sum_{i \in \delta} \left\lceil \frac{i}{2} \right\rceil n^{Di} \\ L^D &= n^{D1} + 2\varphi \\ L^{RD} &= n^{D1} + \varphi \end{aligned}$$

L^R : for Di days $\left\lceil \frac{i}{2} \right\rceil$ exams can be allocated without any *Two-In-a-Row* violation, one exam can be allocated in $D1$ and $D2$ days, and two exams can

be allocated in $D3$ and $D4$ days with an empty period between the two exams.

L^D : one exam can be allocated to a $D1$ day and two exams can be allocated back-to-back to $D2$, $D3$ and $D4$ days without any *Two-In-a-Day* violation.

L^{RD} : we can allocate at most one exam per day without any *Two-In-a-Row* or *Two-In-a-Day* violation.

We have $L^{RD} \leq L^R$, and if there is $D3$ or $D4$ types of days $L^{RD} \leq L^D$.

To compute $\lambda^R(k)$, $\lambda^D(k)$ and $\lambda^{RD}(k)$, that is to say the minimum numbers of edges to be used to count penalties, the idea is to determine in which type of day and period an exam has to be allocated to have the minimum number of violations.

For ***Two-In-a-Row*** soft constraints, if $k \leq L^R$, no violation occurs.

If $L^R < k \leq L^R + n^{D2} + n^{D4}$, we have L^R allocated exams without any violation, one in $D1$ and $D2$ days, two in $D3$ and $D4$ days (empty period between exams), next, each exam allocated to $D2$ or $D4$ days involves one *Two-In-a-Row* violation, so we have $k - L^R$ *Two-In-a-Row* violations. When $k = L^R + n^{D2} + n^{D4}$ exams are allocated, note that there is an empty period in $D3$ and $D4$ days.

If $L^R + n^{D2} + n^{D4} < k$, we have $L^R + n^{D2} + n^{D4}$ allocated exams, next, each exam allocated to $D3$ or $D4$ days involves two *Two-In-a-Row* violations. We have $n^{D2} + n^{D4} + 2(k - (L^R + n^{D2} + n^{D4}))$ violations.

So, the following proposition holds to compute $\lambda^R(k)$:

Proposition 1 *Assume a clique c of size k , we have at least $\lambda^R(k)$ *Two-In-a-Row* violations where $\lambda^R(k)$ is equal to:*

$$\begin{cases} 0 & \text{if } k \leq L^R \\ k - L^R & \text{if } L^R < k \leq L^R + n^{D2} + n^{D4} \\ n^{D2} + n^{D4} + 2(k - (L^R + n^{D2} + n^{D4})) & \text{otherwise} \end{cases}$$

For ***Two-In-a-Day*** soft constraints, if $k \leq L^D$, no violation occurs.

If $L^D < k \leq L^D + n^{D3} + n^{D4}$, we have L^D allocated exams without any violation, we have two back-to-back exams in $D2$, $D3$ and $D4$ days, next, each exam allocated to $D3$ or $D4$ days involves one *Two-In-a-Day* violation (three exams are consecutive). We have $k - L^D$ violations.

If $L^D + n^{D3} + n^{D4} < k$, we have $L^D + n^{D3} + n^{D4}$ allocated exams, only $D4$ days have an empty period, next, each exam allocated involves two *Two-In-a-Day* violations. We have $n^{D3} + n^{D4} + 2(k - (L^D + n^{D3} + n^{D4}))$ violations.

So, the following proposition holds to compute $\lambda^D(k)$:

Proposition 2 *Assume a clique c of size k , we have at least $\lambda^D(k)$ *Two-In-a-Day* violations where $\lambda^D(k)$ is equal to:*

$$\begin{cases} 0 & \text{if } k \leq L^D \\ k - L^D & \text{if } L^D < k \leq L^D + n^{D3} + n^{D4} \\ n^{D3} + n^{D4} + 2(k - (L^D + n^{D3} + n^{D4})) & \text{otherwise} \end{cases}$$

For *Two-In-a-Row* and *Two-In-a-Day* soft constraints, if $k \leq L^{RD}$ no violation occurs.

If $L^{RD} < k \leq L^{RD} + \varphi$, we have L^{RD} exams allocated without any violation, we have one exam per day, next, each exam allocated in $D2$, $D3$, $D4$ days involves a violation (at most φ). We have $k - L^{RD}$ violations.

If $L^{RD} + \varphi < k \leq L^{RD} + \varphi + n^{D3} + n^{D4}$, we have $L^{RD} + \varphi$ allocated exams (two exams in $D2$, $D3$ and $D4$ days), next, each exam allocated in $D3$ or $D4$ days involves two more violations. We have $\varphi + 2(k - (L^{RD} + \varphi))$ violations.

If $L^{RD} + \varphi + n^{D3} + n^{D4} < k$, we have $L^{RD} + \varphi + n^{D3} + n^{D4}$ allocated exams, only $D4$ days have an empty period, next, each exam allocated in $D4$ days involves three violations. We have $\varphi + 2(n^{D3} + n^{D4}) + 3(k - (L^{RD} + \varphi + n^{D3} + n^{D4}))$ violations.

So, the following proposition holds to compute $\lambda^{RD}(k)$:

Proposition 3 Assume a clique c of size k , we have at least $\lambda^{RD}(k)$ *Two-In-a-Row* or *Two-In-a-Day* violations where $\lambda^{RD}(k)$ is equal to:

$$\begin{cases} 0 & \text{if } k \leq L^{RD} \\ k - L^{RD} & \text{if } L^{RD} < k \leq L^{RD} + \varphi \\ \varphi + 2(k - (L^{RD} + \varphi)) & \text{if } L^{RD} + \varphi < k \leq L^{RD} + \varphi + n^{D3} + n^{D4} \\ \varphi + 2(n^{D3} + n^{D4}) \\ + 3(k - (L^{RD} + \varphi + n^{D3} + n^{D4})) & \text{otherwise} \end{cases}$$

Figure 1 provides an illustrative example. On the left, a clique c of $k = 5$ exams, $\{A, B, C, D, E\}$, is displayed with weighted edges (w_{ij}^C are numbers of students taken i and j exams).

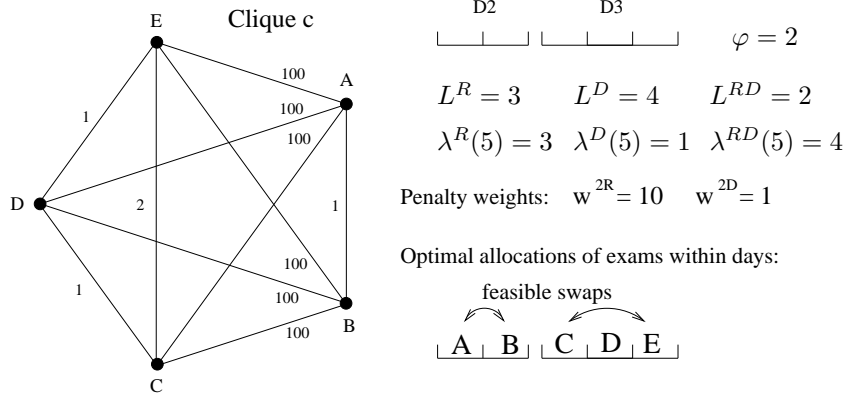


Figure 1: Example of a clique c for which *Two-In-a-Row* or *Two-In-a-Day* soft constraint violations occur.

At the top right of Figure 1, the examination timetabling period has two days, one $D2$ and one $D3$, so $\varphi = 2$. Values for the limits are $(L^R = 3) < k$,

($L^D = 4$) $< k$, and ($L^{RD} = 2$) $< k$, we show that clique c contributes for *Two-In-a-Row* or *Two-In-a-Day* penalties because at least one of these spacing soft constraint violation occurs, so at least one edge has to be used in the computation. However, the numbers of violations of these spacing constraints are not known.

Applying the propositions:

- $\lambda^R(5) = 3 = n^{D2} + n^{D4} + 2(k - (L^R + n^{D2} + n^{D4}))$
- $\lambda^D(5) = 1 = k - L^D$
- $\lambda^{RD}(5) = 4 = \varphi + 2(k - (L^{RD} + \varphi))$

For exams of every clique of size $k = 5$ to be planned within the examination timetabling period three *Two-In-a-Row* violations occur, one *Two-In-a-Day* violation occurs, and four *Two-In-a-Row* or *Two-In-a-Day* violations occur. These values correspond to numbers of weighted edges that contribute to evaluations of corresponding penalties.

3.2 Generic formulation for C^{2R} , or C^{2D} , or $C^{2R} + C^{2D}$ LBs

We propose a generic linear programming formulation to optimally compute lower bounds for *Two-In-a-Row* (C^{2R} term), or *Two-In-a-Day* (C^{2D} term), or *Two-In-a-Row* and *Two-In-a-Day* ($C^{2R} + C^{2D}$) spacing soft constraints. The formulation ensures that an edge (a soft constraint violation) cannot be counted twice whilst taking into account *Coincidence* constraints between exams of cliques.

We denote \mathcal{S} a set of cliques c selected such that ($|c| = k$) is larger than a chosen limit (see Section 3.1) and $\mathcal{E}(\mathcal{S})$ is the set of edges of all cliques in \mathcal{S} .

The idea is to optimally select edges $e = [i, j]$ of cliques $c \in \mathcal{S}$ which can contribute once to lower bounds using $\lambda^R(k)$, $\lambda^D(k)$ and $\lambda^{RD}(k)$, and using for every clique c , the contributions for penalties *Two-In-a-Row*, or *Two-In-a-Day*, or *Two-In-a-Row* and *Two-In-a-Day* denoted as $T^R(c)$, $T^D(c)$ and $T^{RD}(c)$.

To obtain the $T^R(c)$, $T^D(c)$ and $T^{RD}(c)$ optimal values we use a MIP model presented in [7]. Problems to be solved closely resemble set cover problems. All the exams in a clique have to be allocated within days in different periods. Subsets of one, two, three or four exams are used to generate sets of permutations. According to its size a permutation can be allocated to a type of day ($D1$, $D2$, $D3$ and $D4$). Every permutation of a subset of exams can be assessed since *Two-In-a-Row* or *Two-In-a-Day* violations may occur. The model finds an optimal selection of some assessed permutations such that we cover at most all the days and such that all the exams of the clique are allocated.

Not all permutations of a subset of exams need to be generated because many of them have the same penalty induced by soft constraint violations. Breaking symmetries saves many variables but one can further reduce the number of permutations to be used. For the permutations of a subset of exams to be allocated to a type of day a single assessed permutation is retained, since if this

subset of exams is selected to cover one of the days of this type the permutation used has a minimal penalty.

Using the clique c of Figure 1, we illustrate these insights. Let us consider subsets of one, two and three exams to be allocated in $D2$ and $D3$ days.

For every day of type $D2$, and every different exams i and j , we have the following two permutations: (i, j) for exam i in the first period and exam j in the second period, and (j, i) for exam j in the first period and exam i in the second period. These two permutations have the same penalty induced by the *Two-In-a-Row* spacing soft constraint violation. We denote $(i, j) \sim (j, i)$ for “*same induced penalty*”. Permutations with one exam i also need to be considered, since in an optimal solution one exam may be allocated to a $D2$ day, we have: $(i,) \sim (, i)$.

For a $D3$ type of day and one exam i we have: $(i, ,) \sim (, i,) \sim (, , i)$. For two exams i and j we have: $(i, j,) \sim (j, i,) \sim (, i, j) \sim (, j, i)$, and $(i, , j) \sim (j, , i)$. For three exams i, j, k we have: $(i, j, k) \sim (k, j, i)$, and $(i, k, j) \sim (j, k, i)$, and $(j, i, k) \sim (k, i, j)$.

We can further reduce the number of permutations. As an example consider a subset of three exams, assume $T^{RD}(c)$ is to be assessed (*Two-In-a-Row* and *Two-In-a-Day*), and permutations kept are (i, j, k) , (i, k, j) and (j, i, k) . Assume next that in an optimal solution this subset of exams is allocated to a $D3$ day. Since the solution is optimal the permutation used to cover a $D3$ day has a minimum penalty. Therefore, penalties can be computed for permutations (i, j, k) , (i, k, j) and (j, i, k) and the permutation with the smallest penalty retained.

Taking the example of Figure 1, subset of exams $\{C, D, E\}$, and weights $w^{2R} = 10$ and $w^{2D} = 1$. So, (C, D, E) has a penalty of 22, (C, E, D) has a penalty of 31, and (D, C, E) has a penalty of 31. The single permutation to be retained in order to create a decision variable is (C, D, E) .

On left, below, of Figure 1 the optimal allocations of exams are displayed, feasible swaps depicted with curved arrows (symmetries). We obtain $T^R(c) = 30$, $T^D(c) = 2$ and $T^{RD}(c) = 32$, the minimum contributions for penalties.

The generic linear programming formulation we propose to optimally select edges of assessed cliques is the following:

Minimize:

$$\sum_{e=[i,j] \in \mathcal{E}(\mathcal{S})} w_{ij}^C (w^{2R} X_e + w^{2D} Y_e) \quad (1)$$

subject to:

$$\forall e \in \mathcal{E}(\mathcal{S}) \quad X_e + Y_e \leq 1 \quad (2)$$

$$\forall c \in \mathcal{S} \quad \sum_{e=[i,j] \in \mathcal{E}(c)} X_e \geq \lambda^R(c) \quad (3)$$

$$\forall c \in \mathcal{S} \quad \sum_{e=[i,j] \in \varepsilon(c)} Y_e \geq \lambda^D(c) \quad (4)$$

$$\forall c \in \mathcal{S} \quad \sum_{e=[i,j] \in \varepsilon(c)} (X_e + Y_e) \geq \lambda^{RD}(c) \quad (5)$$

$$\forall c \in \mathcal{S} \quad w^{2R} \sum_{e=[i,j] \in \varepsilon(c)} w_{ij}^C X_e \geq T^R(c) \quad (6)$$

$$\forall c \in \mathcal{S} \quad w^{2D} \sum_{e=[i,j] \in \varepsilon(c)} w_{ij}^C Y_e \geq T^D(c) \quad (7)$$

$$\forall c \in \mathcal{S} \quad \sum_{e=[i,j] \in \varepsilon(c)} w_{ij}^C (w^{2R} X_e + w^{2D} Y_e) \geq T^{RD}(c) \quad (8)$$

$$\begin{aligned} \forall c \in \mathcal{S} \quad \forall e = [i, j] \in \varepsilon(c) \quad \text{such that} \quad j \in E^{coin} \\ \forall j' \text{ such that } [j, j'] \in H^{coin} \quad \forall e' = [i, j'] \in \mathcal{E}(\mathcal{S}) \end{aligned} \quad \left\{ \begin{array}{l} X_e = X_{e'} \\ Y_e = Y_{e'} \end{array} \right. \quad (9)$$

$$X_e, Y_e \in \{0, 1\} \quad (10)$$

Decision variable $X_e = 1$ if edge e contributes for a *Two-In-a-Row* violation, zero otherwise, and decision variable $Y_e = 1$ if edge e contributes for a *Two-In-a-Day* violation, zero otherwise.

For all cliques in \mathcal{S} , an edge cannot be counted twice for *Two-in-Row* and *Two-In-a-Day* violations, these disjunctions are enforced using Equations (2).

For every clique, minimum numbers of edges involved are enforced using Equations (3)-(5) (see Section 3.1). For every clique, minimum contributions for penalties are enforced using Equations (6)-(8).

Since endpoints of edges are exams, we can consider *Coincidence* constraints, if there are any, between exams in selected cliques. *Coincidence* constraints are enforced using Equations (9). First consider two exams j and j' subject to a *Coincidence* constraint (i.e. $[j, j'] \in H^{coin}$). Next consider an exam i , an edge $e = [i, j]$, and assume that edge e is selected for one of the spacing violations: edge $e' = [i, j']$ has also to be selected.

Preprocessing proposed in [7] ensures that the two edges $e = [i, j]$ and $e' = [i, j']$ exist by propagating *Coincidence* constraints on the conflict graph $G(E, A_C)$. Note that all edges added by applying this preprocessing have no common students ($w_{ij}^C = 0$) and have no impact in relation to penalties.

Equation (1) assesses the minimum weighted sum of edges $e \in \mathcal{E}(\mathcal{S})$ while respecting for every clique c the minimum numbers of *Two-In-a-Row*, *Two-In-a-Day*, and *Two-In-a-Row* and *Two-In-a-Day* soft constraints violations, the minimum contributions for penalties, and the *Coincidence* constraints between exams of the cliques.

An illustrative example for a conflict graph $G(E, A_C)$ with exams $\{A, B, C, D, E, F, G, H\}$ is shown on top left of Figure 2. We assume weights $w^{2R} = 10$ and $w^{2D} = 1$. We have a common edge $[A, B]$ weighted by $w_{AB}^C = 1$

between cliques c and c' than cannot be counted twice, and a coincidence constraint between exams E and G . For the sake of explanation cliques c and c' are similar, exams $\{F, G, H\}$ play the same role in clique c' as exams $\{C, D, E\}$ in clique c (see also Figure 1).

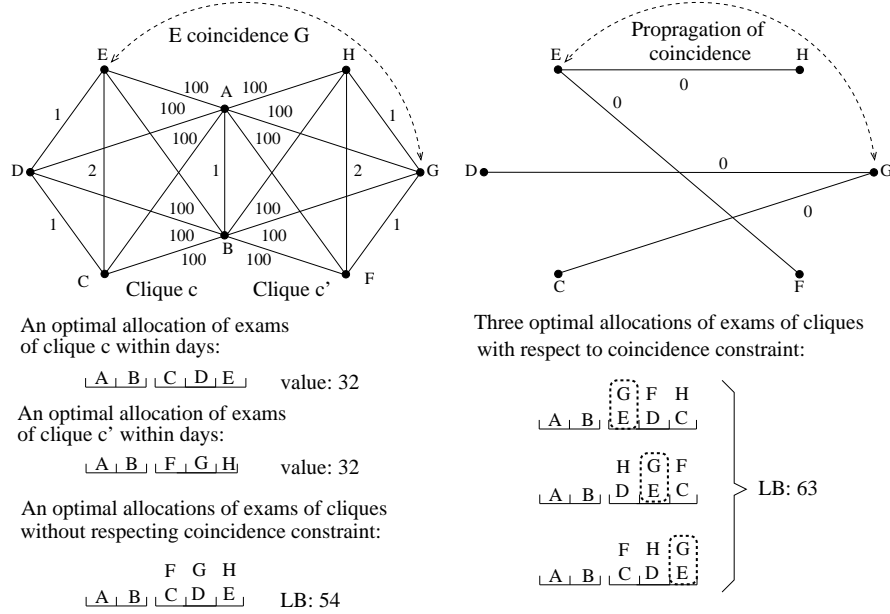


Figure 2: Example for *Two-In-a-Row* and *Two-In-a-Day* lower bound.

On Figure 2, consider the feasible optimal allocations of exams of cliques c and c' (on left, below the conflict graph). The optimal penalty is equal to 32 for both cliques (see also Figure 1).

The greedy heuristic builds a family of edge-disjoint cliques composed of one of the two cliques with a *LB* value of 32 for the *Two-In-a-Row* and *Two-In-a-Day* ($C^{2R} + C^{2D}$) lower bound.

Without considering the coincidence constraint, the formulation computes a lower bound with value of $54 = 10 + 22 + 22$, we have 10 for exams $\{A, B\}$ allocated to $D2$ day, 22 for exams $\{C, D, E\}$ allocated to $D3$, and 22 for exams $\{F, G, H\}$ allocated to $D3$. These values correspond to an optimal allocation of the exams of the two cliques as shown bottom left on Figure 2. We achieve a better lower bound.

With preprocessing, we now consider the coincidence constraint between exams E and G .

At the top right of Figure 2 we display the new edges we add to the conflict graph. For example, edge $[E, H]$ with $w_{EH}^C = 0$ is added to the conflict graph. No student sits the exam E and the exam H , but exams E and G are subject to coincidence constraint, and exams G and H are taken by some students, so

exams E and H cannot be placed in the same period.

Let us consider Equations (9), and focus first on X_e variables associated with a *Two-In-a-Row* soft constraint violation. All equations $X_{[D,E]} = X_{[D,G]}$, $X_{[C,E]} = X_{[C,G]}$, $X_{[G,H]} = X_{[E,H]}$, $X_{[F,G]} = X_{[E,F]}$ must be enforced (edges e and e'). For example, if edge $e = [D, E]$ is used to assess a soft constraint violation ($X_{[D,E]} = 1$), edge $e = [D, G]$ must also be used ($X_{[D,G]} = 1$), this enforces the relative positions of exams $\{D, E, G\}$ considering *Two-In-a-Row* soft constraint violations. Using Equations (9), we also enforce the relative positions of exams considering *Two-In-a-Day* soft constraint violations. By Equations (2) we cannot have $X_e = Y_e$. Hence, the coincidence constraint is enforced without explicitly considering days or periods.

Considering the coincidence constraint, the formulation computes a lower bound with value of $63 = 10 + 22 + 31$. We achieve a better lower bound.

At the bottom right of Figure 2 we display three example of optimal allocations of exams $\{A, B, C, D, E, F, G, H\}$ with respect to the coincidence constraint between exams E and G .

We denote \mathcal{B} the generic formulation that consists of Equations (1) to (10). Numbers of variables are in the order of the number of edges $O(|A_C|)$ and numbers of constraints are in the order of the numbers of edges and cliques $O(|A_C| + |\mathcal{S}|)$.

The proposed generic formulation \mathcal{B} can be used to compute lower bounds either for C^{2R} , or C^{2D} , or $C^{2R} + C^{2D}$. Cliques with common edges can contribute, so no family of edge-disjoint cliques built using a greedy heuristic can provide a better evaluation. The formulation also allows *Coincidence* constraints to be enforced.

4 New formulation of spacing soft constraints

We recall the objective function for ITC2007 examination timetabling problem:

$$\text{Minimize} \quad C^{FL} + C^P + C^R + C^{NMD} + C^{2R} + C^{2D} + C^{PS} \quad (11)$$

where $C^{FL}, C^P, C^R, C^{NMD}, C^{2R}, C^{2D}, C^{PS} \in \mathbb{N}$. These seven terms correspond to *Front-Load*, *Period-Penalty*, *Room-Penalty Non-Mixed-Duration*, *Two-In-a-Row*, *Two-In-a-Day*, and *Period-Spread*.

Although it is able to solve smaller instances, the formulation proposed in [22] requires a large amount of memory. Formulations proposed in [7, 17] aim at reducing the numbers of hard and soft constraints. We denote \mathcal{M} the formulation presented in [7], \mathcal{F}' the formulation proposed by [17] and \mathcal{O} the original formulation described in [22].

In practice it was observed by Fonseca et al. [17] for formulation \mathcal{F}' and Arbaoui et al. [7] for formulation \mathcal{M} that these formulations still faced difficulties to be run on current generation of solvers. The required memory to instantiate sizeable real problems with *Two-In-a-Row*, *Two-In-a-Day*, or *Period-Spread* soft constraints is too large. Formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} have as many *Two-In-*

a-Row, *Two-In-a-Day*, or *Period-Spread* soft constraints and variables as edges in the conflict graph $G(E, A_C)$.

To assess the C^{2R} , C^{2D} and C^{PS} terms, formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} use two kind of equations. The first, based on edges $[i, j] \in A_C$ such that $w_{ij}^C \neq 0$, detects if a violation occurs within days or within a period spread gap g^{PS} , the second count the numbers of students involved.

Based on the work of Glover [18], we propose a more compact formulation for *Two-In-a-Row*, *Two-In-a-Day*, and *Period-Spread* soft constraints.

Considering the conflicting exams with an exam i , i.e. neighbors $\mathcal{N}(i)$ of i in the conflict graph $G(E, A_C)$, the idea is to compute first an upper bound (UB) of the number of students that can be involved if *Two-In-a-Row*, *Two-In-a-Day*, or *Period-Spread* soft constraint violations occur, then, using this UB one can aggregate the assessments of these soft constraints.

For an exam i , the value T_i is an upper bound of the number of students to be counted:

$$\forall i \in E \quad T_i = \sum_{j \in \mathcal{N}(i)} w_{ij}^C \quad (12)$$

At most all the $j \in \mathcal{N}(i)$ exams are allocated in a period p if exam i is not. Assuming exam i is allocated in a period $q \neq p$ such that *Two-In-a-Row*, *Two-In-a-Day*, or *Period-Spread* soft constraint violations occur, T_i is an upper bound of the number of students to be counted.

We denote n^E the number of exams, n^P the number of periods and n^{Day} the number of days.

4.1 Revisiting the Two-In-a-Row soft constraints

For the *Two-In-a-Row* spacing soft constraint, the idea is to assess all the soft constraint violations that occur when an exam i is allocated to a period p of a day and some of its neighbors $j \in \mathcal{N}(i)$ are allocated to the period $p + 1$ of the same day. We propose the following formulation:

$$C^{2R} = w^{2R} \sum_{i \in E} \sum_{p \in P} R_{ip} \quad (13)$$

$$\left. \begin{aligned} \forall i \in E \quad \forall p \in P \quad y_{p(p+1)} = 1 \\ \sum_{[i,j] \in A_C} w_{ij}^C X_{j(p+1)}^P - T_i(1 - X_{ip}^P) \leq R_{ip} \end{aligned} \right\} \quad (14)$$

where boolean decision variables $X_{ip}^P = 1$ iff exam i is in period p , zero otherwise. The variable $R_{ip} \in \mathbb{N}$ assesses the number of students to be counted for the *Two-In-a-Row* penalty if exam i is allocated to period p , and some exams $j \in \mathcal{N}(i)$ are allocated in period $p + 1$ in the same day (parameter $y_{p(p+1)} = 1$).

Considering Equations (14), if some exams $j \in \mathcal{N}(i)$ are allocated in period $p + 1$ we have for these exams $X_{j(p+1)}^P = 1$ and $\sum_{[i,j] \in A_C} w_{ij}^C X_{j(p+1)}^P \leq T_i$. When

$X_{ip}^P = 0$, exam i is not allocated in period p , we have $R_{ip} = 0$ since we minimize C^{2R} term and $R_{ip} \in \mathbb{N}$. When $X_{ip}^P = 1$, exam i is allocated in period p , all Two-In-a-Row penalties are counted for all adjacent exams (i.e. $[i, j] \in A_C$) allocated in period $p + 1$. Hence, Equation (13) counts C^{2R} term.

We now have $O(n^E n^P)$ Equations (13) and (14) compared to $O((n^E)^2 (n^P)^2)$ for formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} (see [7, 17, 22]) using $O(n^E n^P)$ integer variables.

The number of integer variables can be further reduced to $O(n^E)$. An exam i is allocated in a single period p , therefore all R_{ip} are null but one. Equations (13) and (14) can be advantageously rewritten as:

$$C^{2R} = w^{2R} \sum_{i \in E} R_i \quad (15)$$

$$\left. \begin{array}{l} \forall i \in E \quad \forall p \in P \quad y_{p(p+1)} = 1 \\ \sum_{[i,j] \in A_C} w_{ij}^C X_{j(p+1)}^P - T_i(1 - X_{ip}^P) \leq R_i \end{array} \right\} \quad (16)$$

We have one $X_{ip}^P = 1$ and $\sum_{[i,j] \in A_C} w_{ij}^C X_{j(p+1)}^P \leq R_i$, therefore a unique Equation (16) may set $R_i \geq 0$, for all other Equations (16) $X_{iq}^P = 0$ ($p \neq q$), hence all Equations (16) are valid.

For *Two-In-a-Row* soft constraints, formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} require $O((n^E)^2)$ boolean variables, while we have a single integer variable R_i for each exam i . Earlier formulations have $O((n^E)^2 (n^P)^2)$ equations while we now have $O(n^E n^P)$ equations.

4.2 Revisiting the Two-In-a-Day soft constraints

For the *Two-In-a-Day* spacing soft constraint, the idea is to assess all the soft constraint violations that occur when an exam i is allocated to a period p of a day and some of its neighbors $j \in \mathcal{N}(i)$ are allocated to periods $q > p + 1$ of the same day.

We denote $\mathbf{D}_i \in \mathbb{N}$ the variable which assesses the *Two-In-a-Day* penalties for an exam i allocated in a period p and some exams $j \in \mathcal{N}(i)$ allocated in periods $q > p + 1$ such $y_{pq} = 1$ (same day). We propose:

$$C^{2D} = w^{2D} \sum_{i \in E} D_i \quad (17)$$

$$\left. \begin{array}{l} \forall i \in E \quad \forall p \in P \\ \sum_{q \in P, q > p+1, y_{pq}=1} \sum_{[i,j] \in A_C} w_{ij}^C X_{jq}^P - T_i(1 - X_{ip}^P) \leq D_i \end{array} \right\} \quad (18)$$

For Equations (18), if some exams $j \in \mathcal{N}(i)$ are allocated in period $q > p + 1$ such that $y_{pq} = 1$ (same day) we have for these exams $X_{jq}^P = 1$ and $\sum_{q \in P, q > p+1, y_{pq}=1} \sum_{[i,j] \in A_C} w_{ij}^C X_{jq}^P \leq T_i$. When $X_{ip}^P = 0$, exam i is not allocated in period p , we have $D_i = 0$ since the C^{2D} term is minimized and $D_i \in \mathbb{N}$.

When $X_{ip}^P = 1$, exam i is allocated in period p , all Two-In-a-Day penalties are counted for all adjacent exams allocated in periods q of the same day (i.e. $y_{pq} = 1$) but not back-to-back (such that $q > p + 1$) with period p . A single period p is used to allocate exam i , one Equation (18) may set $D_i \geq 0$ and all other Equations (18) are valid. Hence, Equation (17) counts C^{2D} term.

For *Two-In-a-Day* soft constraints, formulations \mathcal{O} and \mathcal{M} require $O((n^E)^2)$ boolean variables, formulation \mathcal{F}' requires $O(n^E n^{Day})$ additional boolean variables, while we have a single integer variable D_i for each exam i . We have $O(n^E n^P)$ Equations (17) and (18) compared to $O((n^E)^2 (n^P)^2)$ Equations for formulations \mathcal{O} and \mathcal{M} . Formulation \mathcal{F}' requires $O(n^E n^{Day})$ additional equations to link the additional boolean variables to boolean decision variables X_{ip}^P and $O((n^E)^2 n^{Day})$ Equations for assessing while the proposed formulation uses $O(n^E n^P)$ Equations (17).

4.3 Revisiting the Period-Spread soft constraints

We present the new formulation of period spread soft constraints for the sake of homogeneity.

For the *Period-Spread* spacing soft constraint, the idea is to assess all the soft constraint violations that occur when an exam i is allocated to a period p and some of its neighbors $j \in \mathcal{N}(i)$ are allocated to periods $q \in \{p+1, p+g^{PS}\}$ within the period spread gap g^{PS} .

We denote $S_i \in \mathbb{N}$ the variable which assesses the *Period-Spread* penalties for an exam i allocated in a period p and some exams $j \in \mathcal{N}(i)$ allocated in periods $q \in \{p+1, p+g^{PS}\}$ (within the period spread gap). We propose:

$$C^{PS} = \sum_{i \in E} S_i \quad (19)$$

$$\left. \begin{array}{l} \forall i \in E \quad \forall p \in P \\ \sum_{q \in \{p+1, p+g^{PS}\}} \sum_{\{i,j\} \in A_C} w_{ij}^C X_{jq}^P - T_i(1 - X_{ip}^P) \leq S_i \end{array} \right\} \quad (20)$$

Considering Equations (20), if some exams $j \in \mathcal{N}(i)$ are allocated in period $q \in \{p+1, p+g^{PS}\}$ we have $X_{jq}^P = 1$ and $\sum_{q \in \{p+1, p+g^{PS}\}} \sum_{\{i,j\} \in A_C} w_{ij}^C X_{jq}^P \leq T_i$. When $X_{ip}^P = 0$, exam i is not allocated in period p , we have $S_i = 0$ since we minimize C^{PS} term and $S_i \in \mathbb{N}$. When $X_{ip}^P = 1$, exam i is allocated in period p , all Period-Spread penalties are counted for all adjacent exams allocated in periods $q \in \{p+1, p+g^{PS}\}$. A single period p is used to allocate exam i , one Equation (20) may set $S_i \geq 0$ and all other Equations (20) are valid. Hence, Equation (19) counts C^{PS} term.

For *Period-Spread* soft constraints, formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} require $O((n^E)^2)$ boolean variables, while we have a single integer variable S_i for each exam i . We have $O(n^E n^P)$ Equations (19) and (20) compared to $O((n^E)^2 (n^P)^2)$ for formulation \mathcal{O} and $O((n^E)^2 n^P)$ for formulations \mathcal{F}' and \mathcal{M} .

4.4 Comparing formulations

We denote \mathcal{L} the formulation that consists of equations of formulation \mathcal{M} , excepted for equations that assess the *Two-In-a-Row*, *Two-In-a-Day*, or *Period-Spread* terms replaced by Equations (15) to (19).

Earlier formulations faced difficulties to be run due to memory overload. Table 1 compares numbers of variables, numbers of **H**ard constraints, and numbers of **S**oft constraints.

Table 1 shows that the proposed formulation \mathcal{L} has less variables and less constraints than existing formulations, so requires less memory to instanciate sizeable real problems.

	\mathcal{O}	\mathcal{F}'	\mathcal{M}	\mathcal{L}
Variables				
<i>Boolean Hard</i>	$n^E n^P$ $+n^E n^R$ $+n^E n^P n^R$	$n^E n^P$ $+n^E n^R$ $+n^E n^P n^R$	$n^E n^P$ $+n^E n^R$ $+n^E n^P n^R$	$n^E n^P$ $+n^E n^R$ $+n^E n^P n^R$
<i>Integer Hard</i>		n^E		
<i>Boolean Soft</i>	$(n^E)^2$ $+n^{Dur} n^P n^R$	$(n^E)^2$ $+n^{Dur} n^P n^R$ $+n^E n^{Day}$	$(n^E)^2$ $+n^{Dur} n^P n^R$	$n^{Dur} n^P n^R$
<i>Integer Soft</i>	$n^P n^R$	$n^P n^R$	$n^P n^R$	$n^P n^R$ $+n^E$
H-Constraints				
<i>Conflict</i>	$n^S n^P$	$n^S n^P$	$n^E n^P$	$n^E n^P$
<i>Link</i>		n^E		
<i>After</i>	$n^{Haft} (n^P)^2$	n^{Haft}	$n^{Eaft} n^P$	$n^{Eaft} n^P$
<i>Coincidence</i>	$n^{Hcoin} n^P$	n^{Hcoin}	$n^{Ecoin} n^P$	$n^{Ecoin} n^P$
<i>Exclusion</i>	$n^{Hexcl} n^P$	$n^{Hexcl} n^P$		
<i>Room Exclusive</i>	$n^E n^{Esole} n^P n^R$	$n^E n^{Esole} n^P n^R$	$n^{Esole} n^P n^R$	$n^{Esole} n^P n^R$
S-Constraints				
C^{NMD}	$n^E n^{Dur} n^P n^R$	$n^E n^{Dur} n^P n^R$	$n^{Dur} n^P n^R$	$n^{Dur} n^P n^R$
C^{2R}	$(n^E)^2 (n^P)^2$	$(n^E)^2 (n^P)^2$	$(n^E)^2 (n^P)^2$	$n^E n^P$
C^{2D}	$(n^E)^2 (n^P)^2$	$(n^E)^2 n^{Day}$ $+n^E n^{Day}$	$(n^E)^2 (n^P)^2$	$n^E n^P$
C^{PS}	$(n^E)^2 (n^P)^2$	$(n^E)^2 n^P$	$(n^E)^2 n^P$	$n^E n^P$

Table 1: Comparing \mathcal{O} , \mathcal{F}' , \mathcal{M} and \mathcal{L} formulations.

We denote n^S the number of students, n^R the number of rooms, n^{Dur} the number of different durations for exams, n^{Haft} the number of pairs $[i, j]$ of exams that correspond to *after* constraints, n^{Eaft} the number of exams subject to *after* constraints, n^{Hcoin} the number of pairs $[i, j]$ of exams that correspond to *coincidence* constraints, n^{Ecoin} the number of exams subject to *coincidence* constraints, n^{Hexcl} the number of pairs $[i, j]$ of exams that correspond to *exclusion* constraints, n^{Esole} the number of exams subject to *room exclusive* constraints.

All the formulations have the same numbers of boolean variables X_{ip}^P (an exam allocated to a period), X_{ir}^R (an exam allocated to a room), X_{ipr}^{PR} (an exam allocated to a period and a room) used to enforce hard constraints.

For *Conflict* constraints (see *H-Constraints*), formulation \mathcal{M} has $O(n^E n^P)$ equations whereas formulations \mathcal{O} and \mathcal{F}' have $O(n^S n^P)$ equations. Generally $n^E \ll n^S$, formulation \mathcal{M} has fewer *Conflict* constraints and we use it for formulation \mathcal{L} .

After and *Coincidence* hard constraints: formulation \mathcal{F}' introduced $O(n^E)$ new integer variables (see *Integer Hard*), this requires $O(n^E)$ additional constraints to link these variables to X_{ip}^P (see *H-Constraints* and *Link*). Formulation \mathcal{O} uses $O(n^{Haft}(n^P)^2)$ and $O(n^{Hcoin}n^P)$ constraints, formulation \mathcal{F}' uses $O(n^{Haft})$ and $O(n^{Hcoin})$ constraints, and formulation \mathcal{M} uses $O(n^{Eaft}n^P)$ and $O(n^{Ecoin}n^P)$ constraints without introducing additional variables. We use equations of formulation \mathcal{M} for formulation \mathcal{L} .

Existing formulations have as many spacing soft constraints as the number of edges in the conflict graph $G(E, A_C)$ for assessing *Two-In-a-Row*, *Two-In-a-Day*, or *Period-Spread* terms. Formulation \mathcal{L} has as many spacing soft constraints as the number of exams. The proposed formulation \mathcal{L} is more compact to instantiate sizeable real-world problems than existing formulations.

5 Results

In this section we present results obtained on the ITC2007 and Yeditepe instances. To make feasible comparisons on lower bounds with earlier results, we consider *Two-In-a-Row* and *Two-In-a-Day* soft constraints ($C^{2R} + C^{2D}$ term), these results were obtained using sets of maximal cliques. We also use the sum of two terms to compare the proposed formulation \mathcal{L} to existing formulations \mathcal{O} , \mathcal{F}' and \mathcal{M} since both terms aim at preventing students sitting more than one exam a day.

Tests were done using a CPLEX 12.5 MIP solver with a single thread and *MipEmphasis* parameter set to *feasibility*, using C++ compiled with gcc version 4.4.7, on a machine with an Intel Xeon E5-2670 and 8 GB of RAM. The computing times are reported in seconds.

Characteristics of ITC2007 and Yeditepe datasets

Table 2 displays characteristics of the ITC2007 and Yeditepe datasets where preprocessing is done. Preprocessing is performed in less than two minutes for each instance (see [7]).

We report instance labels, numbers of exams n^E and numbers of *Coincidence* constraints n^{Hcoin} . Column $|A_C|$ reports numbers of edges of $G(E, A_C)$ conflict graphs, and column ω_{A_C} reports maximum clique sizes. We use the code presented in [26] to compute maximum clique sizes and sets of maximal cliques used to compute lower bounds. Columns L^R , L^D and L^{RD} report the limits of sizes of cliques beyond which at least one violation of *Two-In-a-Row*,

	n^E	n^{Hcoin}	$ A_C $	ω_{A_C}	L^R	L^D	L^{RD}
1	607	2	10308	20	29	-	29
2	870	4	4466	15	24	26	13
3	934	82	13887	21	24	24	12
4	273	4	5792	17	14	14	7
5	1018	19	4890	13	28	28	14
6	242	18	2293	13	8	-	8
7	1096	13	12102	16	40	-	40
8	598	5	9213	17	41	79	40
9	169	2	1193	10	13	-	13
10	214	53	2340	18	22	22	12
11	934	82	13887	21	17	18	9
12	78	2	635	12	7	-	7
yue20011	126	0	1397	14	12	-	12
yue20012	141	0	1792	17	12	-	12
yue20013	26	0	84	6	4	-	4
yue20021	162	0	2320	16	14	-	14
yue20022	182	0	2781	20	14	-	14
yue20023	38	0	145	6	4	-	4
yue20031	174	0	2323	14	12	-	12
yue20032	210	0	3046	16	12	-	12

An LB computable: 15/20

Table 2: Characteristics of ITC2007 and Yeditepe instances. L^R , L^D and L^{RD} limits.

or *Two-In-a-Day*, or *Two-In-a-Row* and *Two-In-a-Day* soft constraint occurs (see Section 3.1).

Instances 1, 6, 7, 9 and 12 have no $D3$ and no $D4$ types of day, so there is no *Two-In-a-Day* soft constraint. All the Yeditepe instances have no *Two-In-a-Day* soft constraints. For these instances, we reported - in column L^D . Instances 1, 5, 7, 8 and 9 are such that ω_{A_C} is less than each limit, so we cannot compute a lower bound.

Instance labels for which a lower bound can be computed on $C^{2R} + C^{2D}$ are shown in bold.

***Two-In-a-Row & Two-In-a-Day* lower bounds**

Computable lower bounds are tabulated in Table 3. Columns $|A_C|$ and $|\mathcal{S}|$ report numbers of edges and sizes of sets of maximal cliques. Column $|A_C| \cdot |\mathcal{S}|$ provides an estimate of the size of the matrix used by model \mathcal{B} since number of variables is of the order of $O(|A_C|)$ and the number of constraints is of the order of $O(|A_C| + |\mathcal{S}|)$ (see Section 3.2). For every sizeable ITC2007 instances, the set of maximal cliques \mathcal{S} is computed in less than two minutes, then, for all selected cliques, we compute $T^R(c)$, $T^D(c)$ and $T^{RD}(c)$ contributions using the MIP summarized in Section 3.2 in less than one hour.

To the best of our knowledge the first work on lower bounds was applied on the first eight ITC2007 instances, column *LB [6]* reports the values for instances 2, 3, 4 and 6 for which $C^{2R} + C^{2D}$ lower bounds can be computed.

To obtain an upper bound on $C^{2R} + C^{2D}$ terms we use UniTime 4.0 [1], while setting to zero the weights for other terms of the objective function. The used approach is an improved version of the work presented by the winner of

	$ A_C $	$ S $	$ A_C \cdot S $	LB [6]	UB_h	LB	t	\mathcal{B}
2	4466	21	92841	10	10	10	10	O
3	13887	853	9732730	330	3300	670	5	O
4	5792	9935	55318080	291	16369	1620	9694	O
6	2293	12638	2268521	1740	5500	2600	70	O
10	2340	8	9064	-	0	0	3	O
11	13887	2262	25809420	-	13890	3970	5821	O
12	635	4501	2493554	-	2613	2030	3	O
yue20011	1397	249	347853	-	52	19	10	O
yue20012	1792	431	772352	-	119	30	18	O
yue20013	84	40	3360	-	29	13	4	O
yue20021	2320	139	322480	-	116	14	15	O
yue20022	2781	151	419931	-	195	34	23	O
yue20023	145	41	5945	-	56	20	6	O
yue20031	2323	187	434401	-	156	24	15	O
yue20032	3046	637	1940302	-	430	74	66	O

Table 3: Results for lower bounds on ITC2007 and Yeditepe instances.

ITC2007 examination timetabling track [25]. Column UB_h reports the values. Columns LB and t report results using formulation \mathcal{B} and computing times.

Optimality is attained by formulation \mathcal{B} for all instances for which a lower bound can be computed (**O** in column t). Excepted for instances 4 and 11 for which $|A_C| \cdot |S|$ values is very large, the computing times for formulation \mathcal{B} are small. For instance 2, optimal value for $C^{2R} + C^{2D}$ terms is obtained. For instance 10, zero is attained for UB , so there exist solutions with no student taking two exams a day.

Formulation \mathcal{B} strickly improves the lower bound by up to 456% (instance 4) compared to results presented in [6]. For all Yeditepe instances, a LB can be quickly computed. Knowing a minimum cost for these spacing soft constraints on days is of practical interest. Instances yue20013 and yue20023 can be optimally solved with costs 29 and 56 respectively (see [27]), we notice that the gaps are 55% and 64%.

Gaps between UB_h and LB are important. Formulation \mathcal{B} optimally selects a set of edges that corresponds to feasible allocations of exams to days without considering capacity constraints. Capacity constraints limit numbers of exams that can be allocated together, thus some exams should have been allocated to other days leading to more *Two-In-a-Row* and *Two-In-a-Day* soft constraint violations.

Examination timetabling problems are hard problems that have been widely investigated in the literature using heuristic approaches. Results on benchmarks are regularly improved applying these solution methods. Approaches for computing lower bounds for *Two-In-a-Row* and *Two-In-a-Day* soft constraints have been investigated less extensively. Bridging the gap between theory and practice remains challenging. As a comparison of columns LB and UB shows, there is still room for improvement.

We proposed an exact method that provides better results than selecting edge-disjoint cliques applying a heuristic. The proposed generic formulation \mathcal{B}

uses minimum numbers of spacing soft constraint violations, and contributions of cliques. Formulation \mathcal{B} ensures that each edge is used once while enforcing coincidence constraints on exams. For spacing soft constraints within days, the formulation allows lower bounds to be computed optimally. for *Two-In-a-Row*, or *Two-In-a-Day*, or *Two-In-a-Row* and *Two-In-a-Day* soft constraints. Better results are achieved compared to earlier proposed lower bounds.

Comparing formulations \mathcal{O} , \mathcal{F}' , \mathcal{M} and \mathcal{L} on $C^{2R} + C^{2D}$

In this section we consider the *Two-In-a-Row* and *Two-In-a-Day* terms of the objective function. Results for formulations \mathcal{O} , \mathcal{F}' , \mathcal{M} and \mathcal{L} are given in Table 4. The first column reports instance labels, and column UB_h reports values running one hour long the code [1]. Columns UB and t show values obtained (UB) and computing times (t) for formulations. A one-hour time limit is used, when reached, - is reported. When no solution can be found, **ns** is reported (no solution). When the eight-gigabyte size limit is reached, **OM**, for out-of-memory, is reported. For formulations \mathcal{O} , \mathcal{F}' , \mathcal{M} and \mathcal{L} , bold is used for best results.

	\mathcal{O}			\mathcal{F}'		\mathcal{M}		\mathcal{L}	
	UB_h	UB	t	UB	t	UB	t	UB	t
1	0	ns	-	ns	-	ns	-	ns	-
2	10		OM	590	-	140	-	235	-
3	3300		OM	ns	-	ns	-	ns	-
4	16369	ns	-	ns	-	ns	-	ns	-
5	0	0	78	135	-	0	77	0	1834
6	5500	5460	-	4910	-	5160	-	4580	-
7	0		OM	0	685	0	687	0	1414
8	0	0	231	0	1246	0	300	0	410
9	0	0	21	0	735	0	17	0	246
10	0	0	22	0	204	0	54	0	242
11	13890		OM	ns	-	ns	-	ns	-
12	2913	4725	-	3185	-	3675	-	3115	-
yue20011	52	158	-	112	-	84	-	73	-
yue20012	119	298	-	184	-	190	-	162	-
yue20013	29	29	5	29	4	29	2	29	3
yue20021	116	182	-	243	-	149	-	125	-
yue20022	195	7046	-	ns	-	396	-	365	-
yue20023	56	56	9	56	-	56	5	56	46
yue20031	156	274	-	316	-	322	-	255	-
yue20032	430	8386	-	657	-	648	-	550	-

Table 4: Comparison of formulations \mathcal{O} , \mathcal{F}' , \mathcal{M} and \mathcal{L} on ITC2007 and Yeditepe instances.

Cliques and **Data-dependent Dual-Feasible Functions** (DDFF) valid inequalities are experienced in [7] and better results were achieved, formulation \mathcal{L} is also run using these valid inequalities.

Formulations \mathcal{F}' and \mathcal{M} proposed in [7, 17] aim at reducing numbers of hard and soft constraints. No out-of-memory (OM) using 8 GB are reported in columns for improved formulations. These formulations can be run on standard computers.

For instances 1, 3, 4 and 11, no solution is attained within the time limit,

regardless of the considered formulation. For instances 5, 7, 8, 9 and 10, all compact formulations obtained solutions with no student taking two exams in the same day (zero cost), using at most half an hour computing time.

For instances 6 and 12 of the ITC datasets, better results are attained using formulation \mathcal{L} . For all Yeditepe instances, optimal solutions or better results are attained using formulation \mathcal{L} compared to others. Formulation \mathcal{M} obtains the best result only for instance 2. Formulation \mathcal{F}' does not provide a solution for the yue20022 instance.

Given the difficulty of timetabling problems, running MIP formulations on sizeable real problems can still be problematic. Nowadays heuristic approaches provide better solutions, as can be seen by looking at Table 4 and comparing column UB_h with the tabulated results. However, comparing the results of heuristic and formulation \mathcal{L} , we have $UB_h = 5500$ and $UB = 4580$ for instance 6. A better result is achieved within the one hour time limit.

Table 4 clearly shows that from formulation \mathcal{O} to formulation \mathcal{L} improvements were obtained. There is however still room for improvements for bridging the gap between exact methods and meta-heuristic approaches.

The proposed formulation \mathcal{L} reduces the number of variables and constraints for *Two-In-a-Row* and *Two-In-a-Day* spacing soft constraints. It is much more compact and can be run on current generations of solvers to provide results within a reasonable computing time.

6 Conclusion

In this paper we proposed improvements to lower bounds and new mathematical formulations for the spacing soft constraints that seek to prevent students sitting more than one exam a day, as defined in the ITC2007 examination timetabling track.

We proposed a generic MIP model to compute lower bounds for Two-In-a-Row and/or Two-In-a-Day spacing soft constraints. The proposed MIP model computes lower bounds by optimally selecting edges of a set of cliques using their individual contributions whilst taking into account exam-coincidence hard constraints. Better lower bounds were achieved than in previous results in the literature.

In existing mathematical formulations of spacing soft constraints the number of equations is in the order of the square of the number of exams. We proposed new formulations of the spacing soft constraints that use as many soft constraints as the number of exams. Our new formulations yield a model that is more compact than existing models. Models are compared with respect to the Two-In-a-Row and Two-In-a-Day terms of the objective function. Using these new formulations, better results were achieved than with existing mathematical formulations.

Although heuristics provide high-quality solutions nowadays, it should be noted that for the benchmark instances considered in this paper, no solution has so far been attained that can also be proved to be optimal (other than for

the two small Yeditepe instances). Some effort still needs to be put into the development of lower bounds and exact methods that are able to validate the quality of heuristics and bridge the gap between theory and practice.

Acknowledgment

This work was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

- [1] University Timetabling, comprehensive academic scheduling solutions. <http://www.unitime.org/>, April, 2018.
- [2] Syariza Abdul-Rahman, Edmund K. Burke, Andrzej Bargiela, Barry McCollum, and Ender Özcan. A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of Operations Research*, 218(1):3–21, 2014.
- [3] Salwani Abdullah, Hamza Turabieh, Barry McCollum, and Paul McMullan. A tabu-based memetic approach for examination timetabling problems. In *Proceedings of the 5th International Conference of Rough Set and Knowledge Technology RSKT 2010, Beijing, China, October 15-17*, volume 6401 of *Lecture Notes in Computer Science*, pages 574–581, 2010.
- [4] Feras Al-Hawari, Mahmoud Al-Ashi, Fares Abawi, and Sahel Alouneh. A practical three-phase ILP approach for solving the examination timetabling problem. *International Transactions in Operational Research*.
- [5] Malek Alzaqebah and Salwani Abdullah. Hybrid bee colony optimization for examination timetabling problems. *Computers & Operations Research*, 54:142–154, 2015.
- [6] Taha Arbaoui, Jean-Paul Boufflet, and Aziz Moukrim. An analysis framework for examination timetabling. In *Proceedings of the Sixth International Symposium on Combinatorial Search (SoCS 2013)*, pages 11–19, July 2013. Leavenworth, WA, USA.
- [7] Taha Arbaoui, Jean-Paul Boufflet, and Aziz Moukrim. Preprocessing and an improved MIP model for examination timetabling. *Annals of Operations Research*, 229:1–22, 2015.
- [8] Edmund K. Burke and Yuri Bykov. The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78, 2017.

- [9] Edmund K. Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society, 64(12):1695–1724, 2013.
- [10] Edmund K. Burke, Kirk Jackson, Jeff Kingston, and Rupert Weare. Automated university timetabling: The state of the art. The Computer Journal, 40(9):565–571, 1997.
- [11] Edmund K. Burke, James P. Newall, and Rupert F. Weare. A memetic algorithm for university exam timetabling. In international conference on the practice and theory of automated timetabling, pages 241–250. Springer, 1995.
- [12] Edmund K. Burke and Sanja Petrovic. Recent research directions in automated timetabling. European Journal of Operational Research, 140(2):266–280, 2002.
- [13] Jacques Carlier, François Clautiaux, and Aziz Moukrim. New reduction procedures and lower bounds for the two-dimensional bin packing problem with fixed orientation. Computers & Operations Research, 34(8):2223–2250, 2007.
- [14] Michael W. Carter and Gilbert Laporte. Recent developments in practical course timetabling. In Proceedings of the Second International Conference of Practice and Theory of Automated Timetabling, volume 1408 of Lecture Notes in Computer Science, pages 3–19. 1998.
- [15] Michael W. Carter, Gilbert Laporte, and Sau Yan Lee. Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society, 47(3):373–383, 1996.
- [16] Meryem Cheraitia and Salim Haddadi. Simulated annealing for the uncapacitated exam scheduling problem. International Journal of Metaheuristics, 5(2):156–170, 2016.
- [17] George H. G. Fonseca and Haroldo G. Santos. A New Integer Linear Programming Formulation to the Examination Timetabling Problem. In The 6th Multidisciplinary International Conference on Scheduling: Theory and Applications (Mista 2013), pages 345–355, August 2013. Gent, Belgium.
- [18] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. Management Science, 22(4):455–460, 1975.
- [19] Christos Gogos, Panayiotis Alefragis, and Efthymios Housos. An Improved Multi-staged Algorithmic Process for the Solution of the Examination Timetabling Problem. Annals of Operations Research, 194:203–221, 2012.

- [20] Graham Kendall and Naimah Mohd Hussin. A tabu search hyper-heuristic approach to the examination timetabling problem at the mara university of technology. In International Conference on the Practice and Theory of Automated Timetabling, pages 270–293. Springer, 2004.
- [21] Barry McCollum, Paul McMullan, Edmund K. Burke, Andrew J. Parkes, and Rong Qu. The Second International Timetabling Competition: Examination Timetabling Track. Technical Report QUB/IEEE/TECH/ITC2007/Exam/v4.0, Queen’s University, Belfast, 2007.
- [22] Barry McCollum, Paul McMullan, Andrew J. Parkes, Edmund K. Burke, and Rong Qu. A New Model for Automated Examination Timetabling. Annals of Operations Research, 194:291–315, 2012.
- [23] Liam T.G. Merlot, Natashia Boland, Barry D. Hughes, and Peter J. Stuckey. A hybrid algorithm for the examination timetabling problem. In International Conference on the Practice and Theory of Automated Timetabling, pages 207–231. Springer, 2002.
- [24] S.A. MirHassani. Improving paper spread in examination timetables using integer programming. Applied Mathematics and Computation, 179(2):702 – 706, 2006.
- [25] Tomás Müller. ITC2007 Solver Description: a Hybrid Approach. Annals of Operations Research, 172:429–446, 2009.
- [26] Patric R.J. Östergård. A fast algorithm for the maximum clique problem. Discrete Applied Mathematics, 120(1–3):197 – 207, 2002.
- [27] Andrew J. Parkes and Ender Özcan. Properties of yeditepe examination timetabling benchmark instances. In Proc. of the 8th international conference on the practice and theory of automated timetabling (PATAT 2010), pages 531–534, August 2010. Queen’s University Belfast, Northern Ireland.
- [28] Rong Qu, Edmund K. Burke, Barry McCollum, Liam T.G. Merlot, and Sau Y Lee. A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling, 12(1):55–89, 2009.
- [29] Rong Qu, Fang He, and Edmund K. Burke. Hybridizing integer programming models with an adaptive decomposition approach for exam timetabling problems. The 4th Multidisciplinary International Scheduling: Theory and Applications, pages 435–446, August 2009. Dublin, Ireland.
- [30] Andrea Schaerf. A survey of automated timetabling. Artificial intelligence review, 13(2):87–127, 1999.

- [31] Amr Soghier and Rong Qu. Adaptive selection of heuristics for assigning time slots and rooms in exam timetables. Applied Intelligence, 39:1–13, 2013.
- [32] Gert Woumans, Liesje De Boeck, Jeroen Beliën, and Stefan Creemers. A column generation approach for solving the examination-timetabling problem. European Journal of Operational Research, 253:178–194, 2016.