An Analysis Framework for Examination Timetabling.

Taha Arbaoui, Jean-Paul Boufflet, Aziz Moukrim

An Analysis Framework for Examination Timetabling

Taha Arbaoui and **Jean-Paul Boufflet** and **Aziz Moukrim**
Université de Technologie de Compiègne
Laboratoire Heudiasyc, UMR 7253 CNRS, 60205 Compiègne, France

Abstract

An examination timetabling problem taken from real world universities was proposed at the International Timetabling Competition (ITC2007). The aim was to establish a common base for comparing different solution approaches. This paper presents new preprocessing methods that disclose hidden constraints and significantly increase the number of new edges that can be added to the *conflict graph*. Results show that the size of the maximum clique of the obtained conflict graph has been more than doubled for two instances as a result of our preprocessing. These larger cliques mean that instances can be analyzed in advance of a solution and end users gain useful information for making decisions. In addition, we have looked at the different criteria that compose the objective function, in order to provide more useful insights into the difficulty of problems in practice. We propose new integer programming formulations using clique inequalities to compute optimal solutions for 4 criteria and to obtain lower bounds for the 3 others. Results are presented and discussed for all the benchmark instances.

1 Introduction

University examination timetabling has been a widely studied problem for a number of decades. The basic problem generally encountered is a graph coloring problem with extra institutional hard and soft constraints. Although the problem appears to be similar across different universities, each individual case is in general unique because of the different rules and terms used by institutions to evaluate the quality of solutions. A wide variety of approaches have been applied to widely differing problem descriptions, and so clear and meaningful scientific comparisons are difficult. The reader should refer to (Qu et al. 2009) for a detailed overview of examination timetabling.

In the second International Timetabling Competition (McCollum et al. 2007), the examination track introduces a real-world formulation encountered within educational institutions. Public instances are at the disposal of the community. The competition constitutes a challenging area, playing an important role in bridging the current gap between research and practice. The definitions of the hard and soft constraints make it possible for a solution to be assessed by measuring the outcomes of heuristic solvers based on different approaches (Gogos, Alefragis, and Housos 2012; McCollum et al. 2009; Müller 2009).

A general integer programming model was proposed (McCollum et al. 2012). This formulation introduces a clear mathematical model, and is useful in evaluating solutions obtained using heuristics. Unfortunately, it cannot be used to find optimal solutions.

A constraint programming approach coupled with an improvement stage was used by the winner of the ITC2007 competition (Müller 2009). The automatic domain reduction mechanism makes use of infeasibility between subproblems when searching for a solution. A partial constructive stage followed by an Extended Great Deluge heuristic was used by (McCollum et al. 2009). In (Gogos, Alefragis, and Housos 2012), the authors used a preprocessing stage to discover hidden dependencies between exams, prior to searching for a solution. They reported that dealing with hard constraints deduced at an early stage has been proved to smooth the solution process. The proposed method then uses a GRASP-based process combined with other meta-heuristic techniques. In a recent work (Demeester et al. 2012), the authors successfully investigated a hyperheuristic approach applied to the instances from their University and to the instances described in (Qu et al. 2009). A hyper-heuristic is a heuristic search method that seeks to automate, the process of selecting, combining, generating or adapting several simpler heuristics to efficiently solve computational search problems. They applied their approach to the second International Timetabling Competition examination track instances, without improving the best solutions that can be found in (McCollum et al. 2009), (Müller 2009) and (Gogos, Alefragis, and Housos 2012).

Since students can sit only one exam at a time, two exams that have some student in common cannot be allocated to the same period. This constitutes the main rule for building the conflict graph between exams. The nodes of the conflict graph are exams, edges are pairs of exam that cannot be assigned to the same period. However, the other hard constraints can be used to discover new *general conflict constraints*, so that two exams cannot necessarily be scheduled together simply because they have no students in common.

We propose three preprocessing stages to exhibit new conflict constraints. As a result, new edges can be added to

the conflict graph in which we are seeking a feasible coloring while respecting the remaining hard constraints that cannot be embedded in the conflict graph. As a consequence, the size of the cliques (i.e. set of pairwise adjacent nodes) may increase, and this improves the efficiency of clique inequalities in an integer programming formulation.

We propose a new formulation for some constraints in the original mathematical model (McCollum et al. 2012). The criteria of the objective function have been considered separately. We optimally solve all the public instances, considering four of these criteria. We then propose an evaluation scheme in order to obtain lower bounds on the three remaining criteria. This means that instances can be analyzed prior to a solution stage, which can help end users to make appropriate decisions.

The remainder of this paper is organized as follows. In Section 2 the problem definitions, the properties of the public instances and the original mathematical formulation (McCollum et al. 2012) are presented. Section 3 describes the proposed preprocessing. New formulations for some constraints of the original model are explained in Section 4. The evaluation scheme for the three remaining criteria is presented in Section 5, and comments about the results are made in Section 6. Our conclusion is to be found in Section 7, where some insights into ongoing work are also given.

2 Problem description

This section provides a brief description of the examination timetabling track of the second International Timetabling Competition (ITC2007). The reader should refer to (McCollum et al. 2007) (McCollum et al. 2010) for a comprehensive presentation. We use the same notation as in (McCollum et al. 2012).

The problem addressed by this track involves allocating a set of exams to a set of rooms within an examination session comprising a fixed number of periods, while satisfying a number of hard constraints. A feasible solution is one in which all hard constraints are respected. The quality of the solution is evaluated using a sum of weighted terms, each term measuring the degree to which some soft constraint is violated.

The benchmark for this competition track consists of 8 public instances and 4 hidden instances used to counter overtuned behavior in competitors' solvers. Each instance is defined as sets of data, hard constraints, and weighted soft constraints. The day, the start time and the duration of each period are given. A set of rooms with individual capacities and weights is also given. Each exam has an individual duration and a set of enrolled students. Each student has to sit a number of exams. In this track, the hard constraints are:

- A student can sit only one exam at a time.
- An exam cannot be split between rooms.
- An exam cannot be split between periods.
- The duration of an exam allocated to a period must be less than or equal to the duration of the period.
- The capacity of individual rooms is not exceeded at any period.
- Exams can share a room, as long as the capacity of the room is respected.
- Precedences, denoted $j \prec i$: exam j has to be scheduled prior to exam i.
- Exclusions, denoted $i \oplus j$: exam i must not take place at the same time as exam j.
- Coincidences, denoted $i \odot j$: the two exams must be assigned to the same period.
- Room exclusives, denoted "room exclusive exam i": exam i must take place in a room on its own.

Seven soft constraints are used as terms of the evaluation function. For each instance, a set of weights is accordingly provided. The terms are defined as follows:

- **Two In a Row:** when two examinations are allocated back to back on the same day, this term corresponds to the number of students that take these exams multiplied by the weight w^{2R}.
- **Two In a Day:** when two examinations are scheduled not back to back but on the same day where there are three periods or more, this term corresponds to the number of students sitting the two exams multiplied by the weight w^{2D}.
- **Period Spread:** this term corresponds to the sum of occurrences of students who have to sit exams within a fixed period spread. The period spread in question is given, and there is no weight.
- **Front Load:** this term corresponds to the number of large exams scheduled in the latter part of the session multiplied by the weight w^{FL}. The number of periods that constitute the "latter part" and the number of candidates that constitute a "large exam" are given.
- **Mixed Duration:** for each period and room, we look at the number of different exam durations allocated. If all the exams have the same duration we count 0, otherwise we count the number of different durations minus one multiplied by the weight w^{NMD}.
- **Period Penalty:** this term corresponds to the number of exams allocated to a penalized period multiplied by its weight w_p^P.
- **Room Penalty:** this term corresponds to the number of exams allocated to a penalized room multiplied by its weight w_r^R.

The first three constraints are an attempt to be as fair as possible to all students taking exams. Since exams with more students enrolled take longer to mark, it is desirable to schedule these exams near the beginning of the examination session (Front Load). The aim of the Mixed-Duration soft constraint is to assign exams which are of equal duration to the same room. Institutions often wish to restrict the use of certain rooms and certain periods to a minimum, and these considerations correspond to the final two soft constraints (Period Penalty and Room Penalty).

In (McCollum et al. 2012) a mathematical formulation is proposed. The authors introduced the conflict graph

$G(E, A_C)$, where E is the set of exams and an edge $[i,j] \in A_C$ if there is at least one student enrolled in exams i and j. An edge $[i,j]$ is weighted by w^C_{ij}, the number of students taking the two exams. P, R and S denote the sets of periods, rooms and students respectively. For the sake of compactness, the original model has been rewritten as:

Minimize:

$$C^{2R} + C^{2D} + C^{PS} + C^{FL} + C^{P} + C^{R} + C^{NMD} \quad (1)$$

Subject to

$$\forall i \in E \quad \sum_{p \in P} X^P_{ip} = 1 \quad (2)$$

$$\forall i \in E \quad \sum_{r \in R} X^R_{ir} = 1 \quad (3)$$

$$\forall i \in E \quad \forall p \in P \quad X^P_{ip} = \sum_{r \in R} X^{PR}_{ipr} \quad (4)$$

$$\forall i \in E \quad \forall r \in R \quad X^R_{ir} = \sum_{p \in P} X^{PR}_{ipr} \quad (5)$$

$$\forall p \in P \quad \forall r \in R \quad \sum_{i \in E} s^E_i X^{PR}_{ipr} \leq s^R_r \quad (6)$$

$$\forall i \in E \quad \forall p \in P \quad d^E_i X^P_{ip} \leq d^P_p \quad (7)$$

$$\forall p \in P \quad \forall s \in S \quad \sum_{i \in E} t_{is} X^P_{ip} \leq 1 \quad (8)$$

$$\left.\begin{array}{r} \forall (i,j) \in H^{aft} \text{ such that } j \prec i \\ \forall p,q \in P \text{ with } p \leq q \qquad X^P_{ip} + X^P_{jq} \leq 1 \end{array}\right\} \quad (9)$$

$$\forall [i,j] \in H^{coin} \quad \forall p \in P \quad X^P_{ip} = X^P_{jp} \quad (10)$$

$$\forall [i,j] \in H^{excl} \quad \forall p \in P \quad X^P_{ip} + X^P_{jp} \leq 1 \quad (11)$$

$$\left.\begin{array}{r} \forall i \in H^{sole} \quad \forall j \in E \quad i \neq j \quad \forall p \in P \\ \forall r \in R \qquad X^P_{ip} + X^R_{ir} + X^P_{jp} + X^R_{jr} \leq 3 \end{array}\right\} \quad (12)$$

$$C^{2R} = w^{2R} \sum_{[i,j] \in A_C} w^C_{ij} C^{2R}_{ij} \quad (13)$$

$$\left.\begin{array}{r} \forall [i,j] \in A_C \quad \forall p \in P \quad \text{with} \quad y_{p(p+1)} = 1 \\ X^P_{ip} + X^P_{j(p+1)} \leq 1 + C^{2R}_{ij} \end{array}\right\} \quad (14)$$

$$C^{2D} = w^{2D} \sum_{[i,j] \in A_C} w^C_{ij} C^{2D}_{ij} \quad (15)$$

$$\left.\begin{array}{r} \forall [i,j] \in A_C \quad \forall p,q \in P \quad |q-p| \geq 2 \quad p < q \\ \text{with} \quad y_{pq} = 1 \qquad X^P_{ip} + X^P_{jq} \leq 1 + C^{2D}_{ij} \end{array}\right\} \quad (16)$$

$$C^{PS} = \sum_{[i,j] \in A_C} w^C_{ij} C^{PS}_{ij} \quad (17)$$

$$\left.\begin{array}{r} \forall [i,j] \in A_C \quad \forall p,q \in P \quad 1 \leq |q-p| \leq g^{PS} \\ p < q \qquad X^P_{ip} + X^P_{jq} \leq 1 + C^{PS}_{ij} \end{array}\right\} \quad (18)$$

$$C^{FL} = w^{FL} \sum_{i \in E^{FL}} \sum_{p \geq g^{FL}} X^P_{ip} \quad (19)$$

$$C^{P} = \sum_{i \in E} \sum_{p \in P} w^p X^P_{ip} \quad (20)$$

$$C^{R} = \sum_{i \in E} \sum_{r \in R} w^r X^R_{ir} \quad (21)$$

$$\left.\begin{array}{r} \forall d \in D \quad \forall i \in E \quad \text{with} \quad u^D_{id} = 1 \quad \forall p \in P \\ \forall r \in R \qquad U^D_{dpr} \geq X^P_{ip} + X^R_{ir} - 1 \end{array}\right\} \quad (22)$$

$$\forall p \in P \quad \forall r \in R \quad 1 + C^{NMD}_{pr} \geq \sum_{d \in D^E} U^D_{dpr} \quad (23)$$

$$C^{NMD}_{pr} \geq 0 \quad (24)$$

$$C^{NMD} = w^{NMD} \sum_{p \in P} \sum_{r \in R} C^{NMD}_{pr} \quad (25)$$

$$X^P_{ip}, X^R_{ir}, X^{PR}_{ipr}, C^{2R}_{ij}, C^{2D}_{ij}, C^{PS}_{ij}, U^D_{dpr} \in \{0,1\} \quad (26)$$

$$\left.\begin{array}{c} C^{2R}, C^{2D}, C^{PS}, C^{FL}, \\ C^{P}, C^{R}, C^{NMD}_{pr}, C^{NMD} \in \mathbb{N} \end{array}\right\} \quad (27)$$

The primary boolean decision variables are X^P_{ip} and X^R_{ir}, $X^P_{ip} = 1$ if exam i is scheduled in period p and $X^R_{ir} = 1$ if exam i is allocated to room r. The secondary boolean variables are: X^{PR}_{ipr}, C^{2R}_{ij}, C^{2D}_{ij}, C^{PS}_{ij} and U^D_{dpr}. The first of these variables is set to 1 if exam i is in period p and room r, 0 otherwise. The following three are used to count the number of soft constraint violations for **Two In a Row**, **Two In a Day** and **Period Spread**. The final variable is used to count of the number of **Mixed Duration** violations. The integer variables C^{2R}, C^{2D}, C^{PS}, C^{FL}, C^{P}, C^{R}, C^{NMD}_{pr} and C^{NMD}, are used to compute the terms of the objective function (see Equation (1)).

Equations (2) and (3) ensure that all the exams are allocated to a period and a room. Equations (4) and (5) link the decision variables.

The room capacities are always respected using Equations (6) in which s^E_i and s^R_r denote respectively the number of students sitting exam i and the seating capacity of room r. The duration hard constraints are respected using Equations (7), in which d^E_i and d^P_p denote the duration of exam i and the duration of period p. The quantities $t_{is} = 1$ if student s is sitting exam i, 0 otherwise. Equations (8) thus enforce the conflict constraints: at any period, any student will be sitting at most one exam.

Sets H^{aft}, H^{coin}, H^{excl} and H^{sole} contain the precedence, coincidence, exclusion and room exclusive constraints. Equations (9), (10), (11), and (12) are used to ensure that the precedence, coincidence, exclusion and room exclusive constraints are respected.

The contributions for the terms C^{2R}, C^{2D}, C^{PS} are set by Equations (13) to (18). $y_{pq} = 1$ in Equations (16) means that periods p and q are on the same day. We therefore count the w^C_{ij} students taking two exams if the two exams i and j are allocated either back to back or on the same day. In the same spirit, Equations (17) and (18) count the C^{PS} term according to the period spread parameter g^{PS}.

The **Front Load** term is computed using (19), where E^{FL} is the set of exams subject to a front load constraint, and g^{FL} is the first period for which a front load penalty has to be counted.

The **Period Penalty** and **Room Penalty** terms are taken into account using Equations (20) and (21).

A U^D_{dpr} decision variable has to be one whenever some exam with duration d uses period p and room r (see (22)). $u^D_{id} = 1$ if exam i has a duration d, 0 otherwise. Equations (23) and (24) count the total number of different durations minus one. Hence, the term C^{NMD} is obtained by applying Equation (25).

3 Preprocessing

Our aim is to deduce as many new edges as possible using the other hard constraints, and then to add these new edges to the conflict graph $G(E, A_C)$ so as to obtain a final general conflict constraints graph $G(E, A_{GC})$.

Let n^P be the number of periods. A feasible solution is an n^P-coloring of the $G(E, A_C)$ graph while respecting the other hard constraints.

A **general conflict constraint** between two exams i and j is defined as a constraint such that these exams cannot be scheduled together in any period. We denote this general conflict constraint as $i \parallel j$.

The initial conflict constraint (students), the precedence constraint $(j \prec i)$ and the exclusion $(i \oplus j)$ constraint all represent general conflict constraints $(i \parallel j)$, since exams i and j cannot be scheduled together in the same period and an edge $[i, j]$ weighted by $w^C_{ij} = 0$ can be added in $G(E, A_C)$.

In (Gogos, Alefragis, and Housos 2012), the authors used a preprocessing stage. They propagated the precedence constraints using the coincidence constraints, as $(j \prec i) \wedge (i \odot k) \Rightarrow (j \prec k)$.

We extend this propagation to any general conflict constraint. Let $(i \parallel j)$ be a general conflict constraint and $[i, j]$ the corresponding edge. Consider two exams i and k subject to a coincidence constraint $(i \odot k)$ and $[j, k] \notin A_C$. The general conflict constraint $(j \parallel k)$ can be deduced, and a new edge $[j, k]$ can therefore be added with $w^C_{jk} = 0$. So, we have: $(j \parallel i) \wedge (i \odot k) \Rightarrow (j \parallel k)$. This propagation is repeatedly applied until no new edges can be deduced: we denote this procedure as $\mathcal{P}$.

Since the coincidence constraints are useful for the propagation, we apply: $(i \odot j) \wedge (i \odot k) \Rightarrow (j \odot k)$ on all the coincidence constraints to exhibit new ones.

To deduce new general conflict constraints we propose using the sizes of exams relative to the room capacities, while respecting the coincidence constraints and the room exclusive constraints. Let i and j be two exams such that $[i, j] \notin A_C$ and set $\mathcal{S}_i$ contains exam i and all exams k such that $(i \odot k)$. Exams in $\mathcal{S}_i$ have to be scheduled in the same period. Set $\mathcal{S}_j$ is similarly built. The objective is to check whether the two sets $\mathcal{S}_i$ and $\mathcal{S}_j$ can be assigned together to the rooms at any period. For this purpose we propose the following model:

maximize:

$$\theta_{ij} = \sum_{r \in R} \sum_{l \in (\mathcal{S}_i \cup \mathcal{S}_j)} X^R_{lr} \tag{28}$$

subject to:

$$\forall l \in (\mathcal{S}_i \cup \mathcal{S}_j) \quad \sum_{r \in R} X^R_{lr} \leq 1 \tag{29}$$

$$\forall r \in R \quad \sum_{l \in (\mathcal{S}_i \cup \mathcal{S}_j)} s^E_l X^R_{lr} \leq s^R_r \tag{30}$$

$$\left.\begin{array}{r} \forall l \in \left(H^{sole} \cap (\mathcal{S}_i \cup \mathcal{S}_j)\right) \quad \forall r \in R \\ \displaystyle\sum_{m \in (\mathcal{S}_i \cup \mathcal{S}_j)\ l \neq m} X^R_{mr} + (\mid \mathcal{S}_i \cup \mathcal{S}_j \mid -1) X^R_{lr} \\ \leq \mid \mathcal{S}_i \cup \mathcal{S}_j \mid -1 \end{array}\right\} \tag{31}$$

$$X^R_{lr} \in \{0, 1\} \tag{32}$$

Recall that R is the set of rooms and s^R_r denotes the capacity of a room r. s^E_l is the number of students sitting exam l, and X^R_{lr} is a boolean decision variable set to one if the exam l is allocated to room r, zero otherwise. H^{sole} is the set of exams subject to a room exclusive constraint.

Equations (29) ensure that an exam $l \in \mathcal{S}_i \cup \mathcal{S}_j$ is allocated at most once to a room. Equations (30) allow the model to enforce the room capacities. The room exclusive constraints for the exams in $\mathcal{S}_i \cup \mathcal{S}_j$ are checked by Equations (31).

If $\theta_{ij} < |\mathcal{S}_i \cup \mathcal{S}_j|$, then exams in $(\mathcal{S}_i \cup \mathcal{S}_j)$ cannot be allocated together in any period p. In this case, a new set of general conflict constraints is discovered: no exam $l \in \mathcal{S}_i$ can be assigned in the same period as an exam $k \in \mathcal{S}_j$ and conversely. We can therefore add an edge for each $l \in \mathcal{S}_i$ and $k \in \mathcal{S}_j$.

Thanks to the proposed preprocessing, we obtain a general conflict graph $G(E, A_{GC})$ where $A_C \subseteq A_{GC}$. A feasible solution is an n^P-coloring of the graph $G(E, A_{GC})$ while respecting the other hard constraints.

4 Revisiting the integer programming formulation

The mathematical model was given in order to formally define the problem. As claimed by the authors (McCollum et al. 2012) this model cannot be used to solve the large competition instances. In a first attempt, we have replaced Equations (8) by $\forall p \in P \quad \forall [i, j] \in A_c \quad X^P_{ip} + X^P_{jp} \leq 1$ in order to reduce the number of conflict constraints. Unfortunately, it does not work for any instance of the competition

even if a single soft constraint is considered at a time. There are too many constraints: this led to problems with memory capacity.

Our objective is to propose modifications that will make it possible to perform computations on the competition instances.

Considering the $G(E, A_{GC})$ graph, $\mathcal{N}(i_1, ..., i_k)$ denotes the neighbours of node i_1 that are also neighbours of node i_2, ..., and also neighbours of node i_k. For instance, $\mathcal{N}(i)$ corresponds to the usual neighbours of node i, and $\mathcal{N}(i,j) = \mathcal{N}(i) \cap \mathcal{N}(j)$ is the set of exams that are neighbours of exams i and j.

Revisiting the general conflict constraint: in (McCollum et al. 2012), the authors used $n^P n^S$ equations that include n^S student parameters to ensure that conflict constraints are respected (see Equation (8) in Section 2). We propose the $n^P n^E$ equations:

$$\left.\begin{array}{l} \forall i \in E \quad \forall p \in P \\ \displaystyle\sum_{j \in \mathcal{N}(i)} X^P_{jp} + \mid \mathcal{N}(i) \mid X^P_{ip} \leq \mid \mathcal{N}(i) \mid \end{array}\right\} \quad (33)$$

that use the $G(E, A_{GC})$ graph. The general conflict constraints are respected: either an exam i or at most $\mathcal{N}(i)$ of its neighbours, i.e. $j \in \mathcal{N}(i)$ in the $G(E, A_{GC})$ graph, can be allocated to a period p.

Revisiting the precedence constraint: the precedence constraints can be grouped as

$$\left.\begin{array}{l} \forall i \in E_{H^{aft}} \quad \forall p \in P \\ \displaystyle\sum_{j \in \mathcal{N}_{aft}(i)} \sum_{q \geq p} X^P_{jq} \\ \qquad + \mid \mathcal{N}_{aft}(i) \mid X^P_{ip} \leq \mid \mathcal{N}_{aft}(i) \mid \end{array}\right\} \quad (34)$$

where $E_{H^{aft}}$ denotes the exams involved in a precedence constraint and $\mathcal{N}_{aft}(i)$ the predecessors $j \in E_{H^{aft}}$ of node i (i.e. $j \prec i$). There are $\mid H^{aft} \mid n^P \times n^P$ equations (9), and now we have $\mid H^{aft} \mid n^P$ equations.

Revisiting the room exclusive constraints: let $i \in H^{sole}$ be an exam that is subject to a room exclusive constraint, and let p be a period and r a room to which exam i can be allocated. The neighbours of exam i, $\mathcal{N}(i)$, cannot be allocated to the same period p. Hence, either the exam i or at most $\mid E \backslash \mathcal{N}(i) \mid$ exams can be allocated to period p and room r. We propose:

$$\left.\begin{array}{l} \forall i \in H^{sole} \quad \forall p \in P \quad \forall r \in R \\ \displaystyle\sum_{k \in (E \backslash \mathcal{N}(i))} X^{PR}_{kpr} \\ \qquad + \mid E \backslash \mathcal{N}(i) \mid X^{PR}_{ipr} \leq \mid E \backslash \mathcal{N}(i) \mid \end{array}\right\} \quad (35)$$

There are $\mid H^{sole} \mid n^P n^R$ Equations (35) while we have $\mid H^{sole} \mid n^E n^P n^R$ Equations (12) in the original model.

Revisiting the mixed duration soft constraint: in (McCollum et al. 2012), the authors used $n^D n^E n^P n^R$ Equations. We denote as E^d the set of exams having duration d. We propose the $(n^D n^P n^R)$ Equations (36) and (37):

$$\left.\begin{array}{l} \forall d \in D \quad \forall p \in P \quad \forall r \in R \\ \qquad |E^d| U^D_{dpr} \geq \displaystyle\sum_{i \in E^d} X^{PR}_{ipr} \end{array}\right\} \quad (36)$$

$$\forall p \in P \quad \forall r \in R \quad 1 + C^{NMD}_{pr} \geq \sum_{d \in D^E} U^D_{dpr} \quad (37)$$

$$C^{NMD}_{pr} \geq 0 \quad (38)$$

$$C^{NMD} = w^{NMD} \sum_{p \in P} \sum_{r \in R} C^{NMD}_{pr} \quad (39)$$

Clique inequality: clique cuts have been successfully applied to a hard timetabling problem by (Avella and Vasil'Ev 2005), who used real-world instances of a university course timetabling problem from an Italian University, and by (Burke et al. 2012) using the instances from Track 3 (i.e the curriculum-based course timetabling problem) of the 2007 International Timetabling Competition. So we introduce:

$$\forall p \in P \quad \forall c \in \mathcal{C} \quad \sum_{i \in c} X^P_{ip} \leq 1 \quad (40)$$

where c is a clique, i an exam in the clique, and $\mathcal{C}$ the set of cliques. The valid clique inequalities are useful in reducing the computation time. We use (Niskanen and Östergård 2003) to compute the maximal cliques for each instance.

We denote $\mathcal{M}$ the modified model that consists of Equations (1) ... (7), (10), (13) ... (21), (26), (27), (33) ... (40).

We use the new formulation to compute successively the optimal values for the C^{FL}_a, C^P_a, C^R_a and C^{NMD}_a terms (subscript a means "alone"). Consequently a new objective function contains only one term, giving us a single criterion. The other soft constraints are not considered. The modifications proposed make it possible to compute the optimal values for each of these terms considered as a single criterion.

5 Lower Bounds for C^{2R}, C^{2D} and C^{PS}

The aim behind the terms Two In a Row, Two In a Day and Period Spread is to place exams for the same student as far apart as possible. For most solutions the major part of the penalty imposed is due to these criteria.

We propose using a set of cliques $\mathcal{C}$ selected according to their sizes. We denote as k the size of a clique c. 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 exist for a particular instance (e.g. instance 2 has $D2$, $D3$ and $D4$ types of day but no $D1$). We denote δ the set of number of periods that corresponds to the types of day of an instance (e.g. for instance 2 we have $\delta = \{2, 3, 4\}$).

We establish the following limits:

Proposition 1. *if $k > L^{2R} = \sum_{i \in \delta} \lceil \frac{i}{2} \rceil n^{Di}$, then we have at least one Two-In-a-Row penalty.*

Proposition 2. *If $k > L^{2D} = n^{D1} + 2 \sum_{i \in (\delta \backslash \{1\})}^{4} n^{Di}$, then we have at least one Two-In-a-Day penalty.*

	n^P	g^{PS}	L^{2R}	L^{2D}	L^{2RD}	L^{PS}
1	54	5	29	ns	29	9
2	40	1	24	26	13	20
3	36	4	24	24	12	8
4	21	2	14	14	7	7
5	42	5	28	28	14	7
6	16	20	8	ns	8	1
7	80	10	40	ns	40	8
8	80	15	41	79	41	5

Table 1: For each instance, n^P: number of periods, g^{PS}: period spread, L^{2R}: limit for two in a row, L^{2D}: limit for two in a day, L^{2RD}: limit for two in a row and two in a day, L^{PS}: limit for period spread.

Proposition 3. *If $k > L^{2RD} = \sum_{i \in \delta} n^{Di}$, then there is at least one penalty due either to Two In a Row, or to Two In a Day.*

Proposition 4. *if $k > L^{PS} = \left\lceil \frac{n^P}{g^{PS}+1} \right\rceil$, then we have at least one Period-Spread penalty.*

All these limits can be stated by counting the number of exams that can be scheduled without the considered soft constraint. Table 1 reports the values for the limits on the size of a clique beyond which a violation of the corresponding soft constraint would necessarily occur. When we have only $D1$ or $D2$ day types, there are no Two-In-a-Day penalties (**ns** in Table 1).

A set of cliques for each instance is built using the code (Niskanen and Östergård 2003). Each clique c is evaluated as described in Sections 5.1 and 5.2.

Assume now that two cliques c and c' have been assessed, if $\mid c \cap c' \mid > 1$ there is at least a common edge: hence their respective costs cannot be added, otherwise an edge may contribute twice.

Proposition 5. *Let $\mathcal{F}$ be a family of cliques embedded in $G(E, A_{GC})$ such that: $\forall c, c' \in \mathcal{F} \quad \mid c \cap c' \mid \leq 1$, therefore $\sum_{c \in \mathcal{F}} C^{2R}(c) + C^{2D}(c) + C^{PS}(c)$ is a lower bound.*

Such a family $\mathcal{F}$ as presented in the above proposition is built using a greedy algorithm. In Section 5.1 we determine the Two-In-a-Row and Two-In-a-Day penalties for a clique, while in Section 5.2 we obtain a lower bound for the Period-Spread penalty.

5.1 C^{2R} and C^{2D} penalties

We now focus on the C^{2R} and C^{2D} lower bound computation. We have $L^{2R} \geq L^{2RD}$, and also $L^{2D} \geq L^{2RD}$ (except when there are no Two-In-a-Day penalties). We therefore consider L^{2RD}: the number of cliques involved in the computation should be potentially larger, as shown by a comparison of the columns L^{2R}, L^{2D} and L^{2RD} in Table 1. The objective function for the modified model $\mathcal{M}$ (see Section 4) is thus to minimize $(C_b^{2R} + C_b^{2D})$ (subscript b means "both"), and the other soft constraints are not considered. As a result, we determine the optimal cost for each clique.

Unfortunately, we observe in practice that the computation time for a clique is of the order of hours even when we use the modified model $\mathcal{M}$, and there are a large number of cliques to evaluate.

We propose a new model that is more effective for computing Two-In-a-Row and Two-In-a-Day penalties per clique. Since we are considering a clique $c \in \mathcal{C}$ of size k, exams are pairwise adjacent: they have to be assigned to k different periods. The examination session extending over n^P periods corresponds to different day types. There are n^{Di} days of type Di with i periods.

The idea is to start by building permutations of the exams in a clique c for each day type comprising the instance ($D1$, $D2$, $D3$, $D4$), and then to optimally select those that can cover all the exams of c with the minimum Two-In-a-Row and Two-In-a-Day cost. There is a large number of permutations. Fortunately, not all the permutations need to be used to find the optimal solution for a clique c. To provide the reader with some insights on how the permutations are selected, let us consider a $D3$ day and a 4-exam clique and focus on two exams $i, j \in c$. There are many feasible permutations of these two exams that can be allocated to the three periods of a $D3$. The periods inside a $D3$ day in which the two exams are scheduled do not matter: only the cost is important. Hence, for a couple of exams i and j a unique permutation with the minimum cost has to be considered. This rationale can be applied on the other types of days.

We evaluate the contribution for a clique c using the following model:

minimize:

$$\sum_{\sigma \in \Pi} C_\sigma X_\sigma \tag{41}$$

subject to:

$$\forall j \in [1, k] \qquad \sum_{\sigma \in \Pi} a_{j\sigma} X_\sigma = 1 \tag{42}$$

$$\forall i \in \delta \quad \sum_{\sigma \in \Pi_i} X_\sigma \leq n^{Di} \tag{43}$$

$$X_\sigma \in \{0, 1\} \tag{44}$$

Subscript σ denotes a permutation. X_σ is a boolean decision variable, equal to 1 if permutation σ is used, 0 otherwise. The permutations are grouped by days, i.e. Π_i is the set of permutations for the days with i periods. Each permutation σ has a cost denoted C_σ that corresponds to its Two-In-a-Row and Two-In-a-Day penalty. Exams j of a k-clique are here numbered from 1 to k. The parameter $a_{j\sigma} = 1$ if exam j belongs to permutation σ, 0 otherwise.

Equations (42) ensure that each exam is assigned exactly once (42). The usage of the different days is enforced by (43). We use at most the associated number of days for each type. Note that the days are anonymous: the model allocates the permutations to an abstract "day", independently of the actual day to which this might be made to correspond in practice. We denote $\mathcal{K}$ the proposed model.

At this stage, the optimal contributions are computed for each clique c in the set $\mathcal{C}$ for which $\mid c \mid > L^{2RD}$.

5.2 C^{PS} penalty

In practice we observe that the time spent finding an optimal solution for a clique c for C^{PS} is longer than the time spent evaluating $(C_b^{2R} + C_b^{2D})$ when we use the proposed model (see Section 4). Moreover, the total number of cliques whose cardinality exceeds L^{PS} is huge.

We propose the following scheme for evaluating the contributions of a clique c.

Assume a clique $\mid c \mid = k > L^{PS}$: there is at least one Period-Spread penalty. First we determine α_k, that is to say the smallest number of terms of the sum used to evaluate C^{PS} that are necessarily involved in the penalty (see Equation (17)). The minimum number of edges α_k for this size of clique k is consequently known, irrespective of clique c (see Equation (18)). Next, to evaluate a clique such that $\mid c \mid = k$, we take the α_k edges that have the smallest penalties.

To compute α_k, the idea is to use the proposed model to find an optimal spacing between k exams over the n^P periods that formally corresponds to an optimal pattern (exams allocated to periods). It is important to remark that for given n^P and g^{PS}, each value α_k is the same for all the cliques such that $\mid c \mid = k$.

We consider formally a clique of size k with a set of anonymous exams $\{i_1, i_2, \cdots, i_k\}$. All the w_{ij} are set to 1. Let us now assume the optimal spacing between the k exams with the minimum value α_k. Since exams are anonymous and $w_{ij} = 1$, two exams can swap places. As a consequence, there are $k!$ feasible permutations. Without lack of generality, we introduce the following total order: $(i_1 \prec i_2 \prec \cdots \prec i_k)$. Any pattern respects this constraint. This total ordering is useful to speed up the computation. Then we use the modified model $\mathcal{M}$ to minimize C^{PS} and the other soft constraints are not considered. Unfortunately it is still time consuming to prove optimality for certain cliques. We stop the computation after ten minutes for each clique.

Before computing the penalties we compute α_k *a priori* for each different clique size such that $k > L^{PS}$.

For each clique, the proposed evaluation that counts the α_k smallest weights over the edges of a clique of size k does not represent the optimal value, but it gives a lower bound for the C^{PS} criterion.

6 Results

All tests are performed using CPLEX 12.5 (IBM 2012), gcc version 4.5.1, on a machine with an Intel Xeon E5430QC@2.66 GHz CPU and 32 GB of RAM.

Table 2 shows the basic properties of the public instances of the examination timetabling track. The density for a $G(E, A)$ graph is computed using $d = \frac{2|A|}{n(n-1)} \times 100$.

The PHC (Period Hard Constraint) column corresponds to the sum of the precedence, coincidence and exclusion constraints, while the RHC (Room Hard Constraint) column shows the number of room exclusive constraints.

In Table 3, the column $room$ shows the number of new edges that result from the preprocessing stage that addresses room capacities in relation to exam sizes.

	d_C	n^E	$\mid A_C \mid$	n^S	n^P	n^R	PHC	RHC
1	5.05	607	9287	7891	54	7	12	0
2	1.17	870	4421	12743	40	49	12	2
3	2.62	934	11410	16439	36	48	170	15
4	15.0	273	5568	5045	21	1	40	0
5	0.87	1018	4500	9253	42	3	27	0
6	6.16	242	1795	7909	16	8	23	0
7	1.93	1096	11595	14676	80	15	28	0
8	4.55	598	8120	7718	80	8	20	1

Table 2: The basic properties of the 8 public instances. d_C: density of the conflict graph, n^E: number of exams, $\mid A_C \mid$: number of edges in $G(E, A_C)$, n^S: number of students, n^P: number of periods, n^R: number of rooms, PHC: number of period hard constraints, RHC: number of room hard constraints

	$room$	$\mathcal{P}$	$\mid A_{GC} \mid$	d_{GC}	t	ω_{A_C}	$\omega_{A_{GC}}$
1	939	81	10308	5.60	17	20	49
2	0	41	4466	1.18	7	15	15
3	0	2477	13887	3.19	64	21	21
4	189	30	5792	15.6	0	17	18
5	0	379	4890	0.94	0	13	13
6	7	487	2293	7.86	1	13	13
7	1	491	12102	2.02	0	16	16
8	906	186	9213	5.16	15	17	48

Table 3: New general conflict constraints deduced when applying the preprocessing stage based on room capacities and exam sizes ($room$). Propagation using coincidence constraints ($\mathcal{P}$). Number of edges ($\mid A_{GC} \mid$) and the density (d_{GC}). Computing times for the preprocessing t, and the overall impact of the preprocessing stages on the maximum clique size (ω_{A_C} relative to $\omega_{A_{GC}}$)

The column $\mathcal{P}$ presents the number of new edges deduced by applying the procedure $\mathcal{P}$ to propagate general conflict constraints using coincidence constraints.

Column $\mid A_{GC} \mid$ shows the number of edges in the new general conflict constraints graph $G(E, A_{GC})$ after applying the preprocessing. Columns ω_{A_C} and $\omega_{A_{GC}}$ give the size of the maximum cliques that were found in $G(E, A_C)$ and $G(E, A_{GC})$ respectively (computed with (Niskanen and Östergård 2003)).

The preprocessing stages based on room capacities and exam sizes are useful in deducing a significant number of general conflict constraints for instances 1, 4 and 8 (see column $room$). A large number of new general conflict constraints are also deduced using the procedure $\mathcal{P}$ on the general conflict constraints. As it can be seen, the computing times are small (0 means less than one second).

Tables 4 and 5 show the optimal values for the frontload (C_a^{FL}), the period penalty (C_a^P), the room penalty (C_a^R) and the non-mixed-duration penalty (C_a^{NMD}). Columns Opt report the optimal solutions found, columns UB report the best value found so far in the literature, and columns t report the computing times in seconds. We provide model $\mathcal{M}$ with initial solutions obtained using the solver presented in (Müller 2009). We set a limit of one day for the computing times.

	C_a^{FL}			C_a^{P}		
	Opt_a	UB	t	Opt_a	UB	t
1	125	255	3042	0	270	1002
2	0	375	8255	0	0	249
3	30	740	2218	0	100	384
4	25	105	6341	50	1750	20648
5	50	1440	657	0	100	42
6	375	375	25	30/55	450	-
7	0	460	250	0	0	339
8	0	380	259	0	342	196

Table 4: Optimal value and best value from the literature for C_a^{FL}: Front Load, C_a^{P}: Period Penalty.

	C_a^{R}			C_a^{NMD}		
	Opt_a	UB	t	Opt_a	UB	t
1	0/350	1050	-	0	100	200
2	0	0	312	0	0	277
3	0	0	305	0	0	126
4	0	0	60	0	0	72
5	0	0	13	0	0	45
6	950	1200	331	0	75	1519
7	0	0	232	0	0	37
8	0	125	463	0	0	41

Table 5: Optimal value and best value from the literature for C_a^{R}: Room Penalty, C_a^{NMD}: Non-Mixed Duration.

In most cases, the end users can gain useful information for making decisions in the scope of the time spent to build a timetable. Further work has to be done to deal with C^R for instance 1 and C^P for instance 6. In these two cases the time limit has been reached: we reported the CPLEX lower bound and the best integer solution.

Not all the soft constraints are of equal importance to the end users, but it can sometimes be useful for end users to know whether a null or lowest-cost solution can be found in relation to each of these criteria. It will be remarked that all the $C_a^{NMD} = 0$, and one can try to search for a solution without Non-mixed-duration violations. The optimal $C_a^{P} = 0$ allows end users to remove the penalized periods prior to the solution process. For Instance 4, there are two large period penalties (200 and 500), and the corresponding periods can also be removed, since a $C_a^{P} = 50$ solution is attainable. In the initial dataset, Room penalties are imposed for instances 1, 2, 3, 6, and 7. The optimal solutions $C_a^{R} = 0$ may be used by end users to avoid all the penalized rooms. The optimal solution can be seen to be close to the best value for Instance 6, and here a large room penalty cannot be avoided. Considering the C_a^{FL} criterion, the value is tightened for Instance 6 ($Opt_a = UB$).

Table 6 displays the values of lower bounds we obtained and the best values from the literature for $(C_b^{2R} + C_b^{2D})$ and C^{PS}. We use (Niskanen and Östergård 2003) to compute all the cliques larger than the considered limits. For Instances 2, 5, 7 and 8 the bounds are tightened for $(C_b^{2R} + C_b^{2D})$. Columns t report the global computing time: clique computing, evaluation of each clique using the model $\mathcal{K}$ and greedy algorithm. Note that for instances 5 and 7, we have

	$(C_b^{2R} + C_b^{2D})$			C^{PS}				
	LB	UB	t	LB	UB	t	LB	Best
1	0	42	5	126	2534	4130	215	4356
2	10	10	79	0	0	0	10	385
3	330	2885	82095	140	4926	4645	501	8996
4	291	9104	52632	47	3925	129	428	15368
5	0	0	nc	142	1259	742	136	2929
6	1740	3700	17223	19900	19900	84	22995	25740
7	0	0	nc	312	3602	3449	220	4037
8	0	0	0	836	6718	4837	552	7461

Table 6: Lower bound and upper bound for $(C_b^{2R} + C_b^{2D})$: Two-In-a-Row + Two-In-a-Day and computing time t in second. C^{PS}: Period Spread and computing time t in second. Sum of the contributions and the best solution found so far in the literature.

$L^{2RD} < \omega_{A_{GC}}$: there is no clique that can be used to compute, the value is zero and we report nc in column t. The evaluation of a clique using the model $\mathcal{K}$ is very fast, but one can have a large number of cliques (e.g. the largest number of cliques is 15598206 for Instance 3). For Instance 6 the best value for C^{PS} is achieved using the lower bound computation: this is the optimal value for this particular instance where $g^{PS} > n^P$ (see (Gogos, Alefragis, and Housos 2012)). The computing times depend on the considered instance for the two lower bounds since they are tightly coupled to the number of cliques larger than the limits.

Penultimate column LB (see Table 6) reports the sum of the contributions of the optimal values and of the obtained lower bounds, while column $Best$ reports the values of the best solutions found so far (see (Gogos, Alefragis, and Housos 2012) (Müller 2009)). As it can be shown these problems remain challenging.

7 Conclusion

We have presented a preprocessing stage that is able to reveal hidden dependencies for the university examination timetabling problem. A significant number of edges can thus be added to the conflict graph. The number of large cliques substantially rises with an increase in density, and new bigger cliques can be exhibited. We have proposed new formulations of constraints that make it possible to compute the optimal value for 4 criteria using a mixed integer program. Lower bounds for the other criteria have been presented. The experimental results were obtained using the public instances of the examination timetabling track of the second International Timetabling Competition (ITC2007). This work constitutes an analysis framework that can potentially help end users make certain decisions prior to a solution.

References

Avella, P., and Vasil'Ev, I. 2005. A Computational Study of a Cutting Plane Algorithm for University Course Timetabling. *Journal of Scheduling* 8(6):497–514.

Burke, E. K.; Mareček, J.; Parkes, A.; and Rudová, H. 2012. A Branch-and-cut Procedure for the Udine Course

Timetabling Problem. *Annals of Operations Research* 194(1):71–87.

Demeester, P.; Bilgin, B.; Causmaecker, P.; and Berghe, G. 2012. A Hyperheuristic Approach to Examination Timetabling Problems: Benchmarks and a New Problem from Practice. *Journal of Scheduling* 15(1):83–103.

Gogos, C.; Alefragis, P.; and Housos, E. 2012. An Improved Multi-staged Algorithmic Process for the Solution of the Examination Timetabling Problem. *Annals of Operations Research* 194:203–221.

IBM. 2012. Cplex User's Manual.

McCollum, B.; McMullan, P.; Burke, E. K.; Parkes, A. J.; and Qu, R. 2007. The Second International Timetabling Competition: Examination Timetabling Track. Technical Report QUB/IEEE/TECH/ITC2007/Exam/v4.0, Queen's University, Belfast.

McCollum, B.; McMullan, P.; Parkes, A.; and Abdullah, S. 2009. An Extended Great Deluge Approach to the Examination Timetabling Problem. In *The 4th Multidisciplinary International Conference on Scheduling: Theory and Applications (Mista 09)*, 424–434. Dublin, Ireland.

McCollum, B.; Schaerf, A.; Paechter, B.; McMullan, P.; Lewis, R.; Parkes, A. J.; Gaspero, L. D.; Qu, R.; and Burke, E. K. 2010. Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition. *INFORMS Journal on Computing* 22(1):120–130.

McCollum, B.; McMullan, P.; Parkes, A.; Burke, K. E.; and Qu, R. 2012. A New Model for Automated Examination Timetabling. *Annals of Operations Research* 194:291–315.

Müller, T. 2009. ITC2007 Solver Description: a Hybrid Approach. *Annals of Operations Research* 172:429–446.

Niskanen, S., and Östergård, P. R. J. 2003. Cliquer User's Guide, Version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland.

Qu, R.; Burke, E.; McCollum, B.; Merlot, L. T. G.; and Lee, S. Y. 2009. A Survey of Search Methodologies and Automated System Development for Examination Timetabling. *Journal of Scheduling* 12(1):55–89.