Computational study of a cutting plane algorithm for University Course Timetabling *

Pasquale Avella † Igor Vasil'ev ‡

July 11, 2003

Abstract

In this paper we describe a successful case-study where a Branch-and-Cut algorithm yields the "optimal" solution of a real-world timetabling problem for university courses *(University Course Timetabling problem)*. The polyhedral structure of the problem is studied in conection with a polytope of the *Set Packing problem*. Several families of cutting planes are introduced, these being crutial to find an optimal solution for the problem presented.

1 Introduction

Given a set of courses, a set of teachers, a set of classes (i.e. a group of students attending exactly the same courses), a set of rooms, a set of time periods and a planning time horizon (usually it is one week), a *School Timetabling problem* consists of assigning courses to rooms and to time periods, satisfying additional constraints that originate from the organization of an educational system.

In order to meet the special requirements arising from different organizational settings of schools, timetabling problems may appear in several forms, thus making it hard to identify a standard formulation. Generally, School Timetabling problems can be classified under three main types:

a) High School Timetabling problem. A set of teachers and set of classes have to be scheduled in such a way that no one teacher (and, respectively, no one class) is involved in more than one lecture at a time. Additional constraints enforce a balancing of workloadfor for both teachers and classes.

*Technical Report 20-03, DIS – Universita di Roma "La Sapienza" (2003)

†RCOST - Research Center on Software Technology, Universitá del Sannio, C.so Garibaldi 107, 82100 Benevento, Italy.

‡Centro di Ricerca in Matematica Pura e Applicata, Università di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy. Institute of System Dynamics and Control Theory, Siberian Branch of Russian Academy of Sciences, Lermontov Srt., 134, 664033 Irkutsk, Russia.

b) **University Course Timetabling problem.** This problem is very similar to that of the High School Timetabling problem above, but in this case, some classes may be joined to enable attendance of some courses. The goal here is to avoid/minimize lecture overlaps with the same students.

c) **Examination Timetabling problem.** The aim of this problem is to reduce student exam overlaps. Where fiasible, exams should be spread out over the whole exam period.

Due to their practical importance, School Timetabling problems have been widely addressed in the literature. We refer the reader to the surveys by Burke and Petrovic [8], Carter [10], Schaerf [20], de Werra [23], and the proceedings of the PATAT conferences [5, 6, 9].

Asratian and de Werra [2], Gotlieb [14] analyzed the basic *class-teacher model* and several its extensions. This model is the core of most real timetabling problems and is defined as follows: given a set of p periods, the problem is to assign each course to some period in such a way that no one teacher (and, respectively, no one class) is involved in more than one lecture at a time. The class-teacher model is formulated as an edge-coloring problem on a bipartite multigraph [23]. It is polynomially solvable, but the addition of some real-life constraints makes the problem NP-complete [24].

School Timetabling is a *feasibility problem* if it aims to find a timetable that meets all the requirements. But often it is transformed into an *optimization problem* by partitioning the constraints into two classes: *hard* constraints and *soft* constraints. The goal is to find a timetable that satisfies all the hard constraints and minimizes violation of the soft constraints.

School Timetabling problems are, computationally, very challenging problems and several solution approaches have been investigated. Local search algorithms have been proposed by Abramson [1], Colorni et al. [11], Corne et al. [12] and Schaerf & Gaspero [19]. Burke et al. [7] have studied genetic algorithms. An extensive overview of other recent heuristic approaches is reported in [5, 6, 8, 9, 20]. A mathematical programming formulation of the School Timetabling problem has been introduced in Tripathy [21]. Birbas et al. [3] presented the formulation of a timetabling problem for Greek High Schools.

In this paper we study the polyhedral structure of the *University Course Timetabling problem* to provide effective classes of cutting planes. We describe a Branch-and-Cut algorithm to solve a real-world instance for a Faculty with 69 courses, 59 teachers and 15 rooms.

The problem is formulated as a *Set Packing problem* with side constraints. To tighten the initial formulation we introduce well-known families of valid inequalities of the Set Packing problem polytope, namely, Clique and Lifted Odd-Hole inequalities. Further we introduce new families of valid inequalities, which do not derive themselves from the analysis of a Set Packing polytope.

The remainder of the paper is organized as follows. In section 2 we define the problem. In section 3 we present the Integer Programming formulation. In section 4 we show the relation to the Set Packing problem and we also describe separation algorithms for Clique and Odd-Hole inequalities. In section 5, new families of valid inequalities and their related separation algorithms are presented. Finally, in section 6, we show the effectiveness of the algorithm by reporting on computational experience in a real-world case-study.

2 Problem statement

In this section, we introduce some notations and definitions. We will then describe the problem requirements.

- Let $C = \{1, \ldots, \bar{c}\}$ be a set of courses. For any $c \in C$, let n_c be the number of hours to be scheduled per week and let $n^c_{\min}$ and $n^c_{\max}$ be, respectively, the minimum and maximum daily number of teaching hours (no less than $n^c_{\min}$ and no more than $n^c_{\max}$ hours have to be assigned to the day d, if the course c is scheduled in d).
- Let $R = \{1, \ldots, \bar{r}\}$ be a set of rooms.
- Let $T = \{1, \ldots, \bar{t}\}$ be a set of time periods (of the same length, one hour).
- Let $D = \{1, \ldots, \bar{d}\}$ be the days of the week when teaching is allowed. For any $d \in D$, let τ_d and ι_d be, respectively, the first time slots of the morning and afternoon session in day d.
- Let $G = \{1, \ldots, \bar{g}\}$ be a set of classes (classes are groups of students attending exactly the same courses). For any class $g \in G$, let $C_g \subset C$ denote courses that the class g should attend. In the contrast of the School Timetabling problem, it can happen that $C_{g_1} \cap C_{g_2} \neq \emptyset$, i.e. some classes can attend the same courses.
- Let $S = \{1, \ldots, \bar{s}\}$ be a set of teachers and, for any $s \in S$, let $\mathcal{C}_s \subset C$ be a subset of courses taught by teacher s. Let k_s denote the maximum weekly number of teaching days for the teacher s.
- Let $l_{\max}$ be the maximum daily number of teaching hours for any class $g \in G$.
- Let p_{ct} be a penalty occurring if the course c is scheduled at the time t. Usually p_{ct} measures the "undesirability" of the corresponding teacher s ($c \in \mathcal{C}_s$) to teach the course c at time t.

A *timetable* is an assignment of courses C to rooms R and to time periods T. A timetable should satisfy the following standard requirements, requirements common to most of the educational systems.

i) For each course $c \in C$: n_c hours a week must be scheduled.

ii) For each class $g \in G$: class g cannot attend more than one course at time $t \in T$.

iii) For each teacher $s \in S$: teacher s cannot teach more than one course at time $t \in T$.

iv) For each room $r \in R$: room r cannot host more than one course at time $t \in T$.

In addition to the basic requirements, the timetable should meet the following "local" requirements, charaterizing the real-world case study we have considered:

v) If a course $c \in C$ is scheduled in day $d \in D$, it should take between $n^c_{\min}$ and $n^c_{\max}$ hours.

vi) For each course $c \in C$, the timetable should be "compact". If two hours of the same course c are scheduled in day d, they have to be assigned to adjacent time periods. In other words, let t_1 and t_3 ($t_1 < t_3$) be time periods belonging to the same day. If course c is assigned to the time periods t_1 and t_3, the same course should be scheduled at every time period between t_1 and t_3 as well, to guarantee the adjacency.

vii) All the hours of a course $c \in C$ scheduled in a $d \in D$ should be located in the same room $r \in R$.

viii) No class can attend more than l_{max} teaching hours a day.

ix) We distinguish two sessions (morning and afternoon). Courses of a class can be taught only either in the morning or in the afternoon session.

x) For each class $g \in G$, the timetable should be "compact": for each class, empty periods between two of its courses are not allowed.

xi) A teacher $s \in S$ cannot work for more than k_s days a week.

xii) Due to the availability of equipments and capacity of the room, course $c \in C$ can be assigned to a subset of rooms $R_c \subseteq R$.

xiii) A room $r \in R$ is available in a subset of time slots $T_r \subseteq T$.

xiv) A teacher $s \in S$ is available in a subset of time slots $\mathcal{T}_s \subseteq T$.

xv) The sum of the penalties should be minimized.

3 Integer Linear Programming formulation

To define an Integer Programming formulation for the University Course Timetabling problem, we must introduce three sets of binary variables:

- x_{crt} is 1 if course $c \in C$ is scheduled in room $r \in R$ at time $t \in T$, $x_{crt} = 0$ otherwise;
- u_{cd} is 1 if course $c \in C$ is assigned to the day $d \in D$, 0 otherwise;
- ψ_{sd} is 1 if $d \in D$ is a teaching day for teacher $s \in S$, 0 otherwise.

With these variables, a formulation that meets both basic and local requirements is:

$$\min \sum_{c \in C} \sum_{t \in T} p_{ct} \sum_{r \in R} x_{crt} \tag{1}$$

$$\sum_{r \in R} \sum_{t \in T} x_{crt} = n_c, \qquad c \in C \tag{2}$$

$$\sum_{c \in C_g} \sum_{r \in R} x_{crt} \leq 1, \qquad g \in G,\ t \in T \tag{3}$$

$$\sum_{c \in \mathcal{C}_s} \sum_{r \in R} x_{crt} \leq 1, \qquad s \in S,\ t \in T \tag{4}$$

$$\sum_{c \in C} x_{crt} \leq 1, \quad r \in R, t \in T \tag{5}$$

$$\sum_{r \in R} \sum_{\tau_d \leq t < \tau_{d+1}} x_{crt} \geq n^c_{\min} u_{cd}, \quad c \in C, d \in D \tag{6}$$

$$\sum_{r \in R} \sum_{\tau_d \leq t < \tau_{d+1}} x_{crt} \leq n^c_{\max} u_{cd}, \quad c \in C, d \in D \tag{7}$$

$$\sum_{r \in R} (x_{crt_1} - x_{crt_2} + x_{crt_3}) \leq 1, \qquad \begin{array}{l} c \in C,\ d \in D, \\ \tau_d \leq t_1 < t_2 < t_3 < \tau_{d+1} \end{array} \tag{8}$$

$$x_{cr_1t_1} + x_{cr_2t_2} \leq 1, \qquad \begin{array}{l} c \in C,\ 1 \leq r_1 < r_2 \leq \bar{r}, \\ d \in D, \tau_d \leq t_1 < t_2 < \tau_{d+1} \end{array} \tag{9}$$

$$\sum_{c \in C_g} \sum_{r \in R} \sum_{\tau_d \leq t < \tau_{d+1}} x_{crt} \leq l_{\max}, \quad g \in G, d \in D \tag{10}$$

$$\sum_{r \in R} x_{c_1rt_1} + \sum_{r \in R} x_{c_2rt_2} \leq 1, \qquad \begin{array}{l} g \in G,\ c_1, c_2 \in C_g,\ c_1 \neq c_2,\ d \in D, \\ \tau_d \leq t_1 < \iota_d \leq t_2 < \tau_{d+1} \end{array} \tag{11}$$

$$\sum_{c\in C_g}\sum_{r\in R}(x_{crt_1}-x_{crt_2}+x_{crt_3})\le 1, \quad \begin{array}{l} g\in G,\ d\in D,\\ \tau_d\le t_1<t_2<t_3<\tau_{d+1}\end{array} \tag{12}$$

$$\sum_{r\in R} x_{crt}\le \psi_{sd}, \quad c\in\mathcal{C}_s,\ s\in S,\ d\in D,\ \tau_d\le t<\tau_{d+1} \tag{13}$$

$$\sum_{d\in D}\psi_{sd}\le k_s, \quad s\in S \tag{14}$$

$$\begin{array}{ll} x_{crt}\in\{0,\ 1\}, & c\in C,\ r\in R,\ t\in T\\ u_{cd}\in\{0,\ 1\}, & c\in C,\ d\in D\\ \psi_{sd}\in\{0,\ 1\}, & s\in S,\ d\in D\end{array}$$

Objective function (1) attempts to minimize the sum of penalties (requirement xv).

Constraints (2) enforce that n_c is the number of weekly hours for each course c (requirement i)). Constraints (3) impose that a class g cannot attend more than one course at time t (requirement ii)). Requirement iii) – a teacher cannot teach more than one course at time t – is enforced by constraints (4). Requirement iv) – a room r cannot host more than one course at time t – is enforced by constraints (5).

Constraints (6) and (7) impose that, if course c is scheduled in day d, i.e. if $u_{cd}=1$, the number of daily hours of course c is between n^c_{min} and n^c_{max} (requirement v)). Constraints (8) guarantee compactness for course c, the time periods assigned to day d should be adjacent (requirement vi)). Constraints (9) impose that all hours of course c scheduled in the same day d, be assigned to the same room r (requirement vii)).

Constraints (10) impose an upper bound $l_{\max}$ of the number of daily teaching hours that a class has to attend (requirement viii)). Constraints (11) do not allow a class to attend courses both in the morning and afternoon sessions of the same day (requirement ix)). For example, if class g attends course c_1 in the morning session of day d ($x_{c_1rt}=1$ for some t such that $\tau_d\le t<\iota_d$), class g cannot attend other courses in the afternoon session ($x_{c_2rt}=0$ for all $c_2\in C_g$ and $\iota_d\le t<\tau_{d+1}$). Constraints (12) guarantee that the timetable for a class g is compact (requirement x)).

Finally, constraints (13), (14) limit the number of working days for each teacher (requirement xi)).

For clarity, we do not consider requirements concerning capability/availability of rooms and teachers (requirements xii)-xiv)). Such requirements can be easily expressed by fixing corresponding variables $x_{crt}=0$ and removing them from the formulation, i.e

$$x_{crt}=0, \quad c\in C, r\in R\setminus R_c,\ t\in T \tag{15}$$

$$x_{crt}=0, \quad c\in C, r\in R,\ t\in T\setminus T_r \tag{16}$$

$$x_{crt}=0, \quad s\in S, c\in\mathcal{C}_s,\ r\in R,\ t\in T\setminus\mathcal{T}_s \tag{17}$$

We denote by P_{UCTP} the University Course Timetabling polytope, i.e. the convex hull of integer solutions of formulation (1)-(17).

4 A Set Packing relaxation

Here we investigate the relation of the University Course Timetabling problem to the Set Packing problem. This relation will be used to derive valid inequalities for P_{UCTP} from previously known results for the Stable Set polytope [4, 22].

Let $G(V, E)$ be a graph with node weights w. The *Set Packing* or *Stable Set problem* (SSP) is to find a set of pairwise non-adjacent nodes of maximum weights, or, equivalently, solve the integer linear problem

$$\max c^T y, \quad Ay \leq 1,\ y \in I\!B^{|V|}$$

where binary variables y_i, $i \in V$, correspond to each node. The matrix A is an edge-node (or, more strictly, clique-node) incidence matrix, or, vice-versa, the graph is a *conflict* or *column intersection* graph of matrix A.

The following subset of constraints (3)-(5), (9), (11) of the formulation, define a *Set Packing relaxation* of the University Course Timetabling problem:

$$\begin{array}{ll}
\sum\limits_{c\in C_g}\sum\limits_{r\in R} x_{crt} \leq 1, & g \in G, t \in T \\
\sum\limits_{c\in \mathcal{C}_s}\sum\limits_{r\in R} x_{crt} \leq 1, & s \in S,\ t \in T \\
\sum\limits_{c\in C} x_{crt} \leq 1, & r \in R,\ t \in T \\
x_{cr_1t_1} + x_{cr_2t_2} \leq 1, & \begin{array}{l} c \in C,\ 1 \leq r_1 < r_2 \leq \bar{r}, \\ d \in D,\ \tau_d \leq t_1 < t_2 < \tau_{d+1} \end{array} \\
\sum\limits_{r\in R} x_{c_1rt_1} + \sum\limits_{r\in R} x_{c_2rt_2} \leq 1, & \begin{array}{l} g \in G,\ c_1, c_2 \in C_g,\ c_1 \neq c_2,\ d \in D, \\ \tau_d \leq t_1 < \iota_d \leq t_2 < \tau_{d+1} \end{array}
\end{array}$$

After complementing variables $\bar{u}_{cd} = 1 - u_{cd}$, $c \in C$, $d \in D$, and $\bar{\psi}_{sd} = 1 - \psi_{sd}$, $s \in S$, $d \in D$, we can strengthen the Set Packing formulation by adding the following inequalities, which can easily be proved to be valid for P_{UCTP}:

$$\sum_{c\in \mathcal{C}_s}\sum_{r\in R} x_{crt} + \bar{\psi}_{cd} \leq 1, \quad s \in S, d \in D, \tau_d \leq t < \tau_{d+1} \tag{18}$$

$$x_{cr_1t_1} + x_{cr_2t_2} + \bar{u}_{cd} \leq 1, \quad \begin{array}{l} c \in C,\ 1 \leq r_1 < r_2 \leq \bar{r}, \\ d \in D,\ \tau_d \leq t_1 < t_2 < \tau_{d+1} \end{array} \tag{19}$$

We observe that inequalities (18) dominate inequalities (4), (13), and that inequalities (19) dominate inequalities (9).

The following family of inequalities arises by combining inequalities (10) with class compactness inequalities (12).

Proposition 4.1 *Inequalites*

$$\sum_{c\in C_g}\sum_{r\in R}(x_{crt_1} + x_{crt_2}) \leq 1, \quad \begin{array}{l} g \in G,\ d \in D, \\ \tau_d \leq t_1 + l_{\max} \leq t_2 < \tau_{d+1} \end{array} \tag{20}$$

are valid for P_{UCTP} and imply inequalities (10).

Proof. Let $g \in G$ and let $c_1, c_2 \in C_g$. Let $d \in D$, t_1 and t_2 such that $\tau_d \leq t_1 + l_{\max} \leq t_2 < \tau_{d+1}$. Suppose that $x_{c_1rt_1} = 1$ and $x_{c_2rt_2} = 1$ for some $r_1,\ r_2 \in R$, so that inequality (20) is violated. By compactness constraints (12), it follows that $\sum_{c \in C_g} \sum_{r \in R} x_{crt_3} = 1$, for each t_3 such that $t_1 < t_3 < t_2$. But then we have more that $l_{\max}$ teaching hours for class g in day d, so violating constraints (10), a contradiction. $\diamond$

The following inequalities arise by combining constraints (7), (8) and (11).

Proposition 4.2 *Let*

$$T_{cdt_1} = \{t = t_1 + kn^c_{\max} :\ k \in I\!N \cup \{0\},\ \tau_d \leq t < \iota_d\}$$

$$\mathcal{T}_{cdt_2} = \{t = t_1 + kn^c_{\max} :\ k \in I\!N \cup \{0\}, \iota_d \leq t < \tau_{d+1}\}$$

Periodic conflict inequalities

$$\sum_{r \in R} \sum_{t \in T_{c_1 dt_1}} x_{c_1rt} + \sum_{r \in R} \sum_{t \in \mathcal{T}_{c_2 dt_2}} x_{c_2rt} \leq 1, \qquad \begin{array}{l} c_1, c_2 \in C, d \in D, \\ \tau_d \leq t_1 < \tau_d + n^{c_1}_{\max}, \\ \iota_d \leq t_2 < \iota_d + n^{c_2}_{\max}, \end{array} \tag{21}$$

are valid for P_{UCTP} and dominate inequalities (11) and imply (7).

Proof. First, we prove that

$$\sum_{r \in R} \sum_{t \in T_{c_1 dt_1}} x_{c_1rt} \leq 1, \quad d \in D, \tau_d \leq t_1 < \tau_d + n^{c_1}_{\max}$$

Let $t_2, t_3 \in T_{c_1dt_1}$ and suppose that $x_{c_1rt_2} = 1$ and $x_{c_1rt_3} = 1$ for room r (we are considering one room due to constraint (9)), so the inequality is violated. According to compactness inequalities (8), $x_{c_1rt} = 1$ for all $t_2 < t < t_3$, as soon as $|t_2 - t_3| \geq n^{c_1}_{\max}$ by definition, it contradicts that the number of hours a day cannot be more than $n^{c_1}_{\max}$.

Analogously, we can prove that

$$\sum_{r \in R} \sum_{t \in \mathcal{T}_{c_2 dt_2}} x_{c_2rt} \leq 1, \quad d \in D, \iota_d \leq t_2 < \iota_d + n^{c_2}_{\max}$$

By constraints (11), we have that $x_{c_1rt_1} + x_{c_2rt_2} \leq 1$, for each c_1, c_2 form C_g and t_1, t_2 of the different sessions and the proof is complete. $\diamond$

As a final result, we find that the strengthened Set Packing relaxation of the timetabling problem is defined by constraints (3), (5), (18)–(21). We can build the intersection graph associated with this problem, by assigning a node to each variable and an edge for two variables (nodes), which share the same constraints.

We use Set Packing relaxation as a logical framework to derive two well-known families of cutting planes, namely Clique and Lifted Odd-Hole inequalities [15], that turn out to be very effective at tightening of formulation.

For separation of Clique inequalities we adopt an "exact" version of Maximum Clique identification procedure by Hoffman and Padberg [15] (we refer to apendix A for further details on this procedure).

Violated Odd-hole inequalities can be identified in polynomial time [17]. Here, because of the large size of instances, we apply the fast heuristic proposed by Hoffman and Padberg [15] (we refer to appendix B for further details). Odd-hole inequalities can be strengthened by sequential lifting. We compute the lifting coefficients by the fast Mannino and Sassano algorithm for Maximum Cardinality Stable Set problem [18].

5 Cutting planes

In this section we introduce five families of valid inequalities for P_{UCTP}.

Proposition 5.1 *Let* $\hat{C} = \{c \in C : n^c_{\min} \geq 2\}$. *Adjacency inequalities*

$$-x_{crt-1} + x_{crt} - x_{crt+1} \leq 0, \qquad \begin{array}{l} c \in \hat{C}, r \in R, d \in D, \\ t \in [\tau_d + 1,\ \iota_d - 2] \cup [\iota_d + 1,\ \tau_{d+1} - 2], \end{array} \tag{22}$$

$$x_{crt} - x_{crt+1} \leq 0, \qquad c \in \hat{C},\ r \in R,\ d \in D,\ t = \tau_d, \iota_d, \tag{23}$$

$$-x_{crt-1} + x_{crt} \leq 0, \qquad c \in \hat{C},\ r \in R, d \in D,\ t = \iota_d - 1, \tau_{d+1} - 1 \tag{24}$$

are valid for P_{UCTP}.

Proof. If $x_{crt} = 0$, inequalities (22)-(24) are trivially valid.

- Let $c \in \hat{C}$ and let $t \in [\tau_d + 1,\ \iota_d - 2] \cup [\iota_d + 1, \tau_{d+1} - 2]$. If $x_{crt} = 1$, by compactness constraints (8), we have that course c must be assigned at least to one of two adjacent time periods $(t-1)$ or $(t+1)$, so that $x_{crt-1} + x_{crt+1} \geq 1$ and inequality (22) is valid.

- Let $t = \tau_d$ or ι_d. If $x_{crt} = 1$, by compactness constraints (8), we have that course c must be assigned to adjacent time period $(t+1)$, so that $x_{crt+1} = 1$ and inequality (23) is valid.

- Let $t = \iota_d - 1$ or $\tau_d - 1$. If $x_{crt} = 1$, by compactness constraints (8), we have that course c must be assigned to adjacent time period $(t-1)$, so that $x_{crt-1} = 1$ and inequality (24) is valid.

◇

We are given the minimum and maximum daily number of hours allowed for a course c, so we can compute, respectively, the minimum and maximum number of days when the course can take place.

Proposition 5.2 *Min (Max) Busy Days inequalities*

$$\begin{array}{l} \sum\limits_{d\in D} u_{cd} \leq \left\lfloor \frac{n_c}{n^c_{\min}} \right\rfloor, \quad c \in C \\ \sum\limits_{d\in D} u_{cd} \geq \left\lceil \frac{n_c}{n^c_{\max}} \right\rceil, \quad c \in C \end{array} \tag{25}$$

are valid for P_{UCTP}.

Proof. According to (2), (6) and (7), we have

$$\begin{array}{l} n^c_{\min} \sum\limits_{d\in D} u_{cd} \leq n_c, \quad c \in C \\ n^c_{\max} \sum\limits_{d\in D} u_{cd} \geq n_c, \quad c \in C \end{array}$$

Then we divide both sides by $n^c_{\min}$ (respectively, $n^c_{\max}$) and round down (respectively, round up) the right hand side. ◇

Proposition 5.3 *Let*

$$\hat{C} = \{c \in C : \ n^c_{\min} = n^c_{\max}\}$$

$$T_{cdt_1} = \{t = t_1 + kn^c_{\max} : \ k \in I\!N \cup \{0\}, \ \tau_d \leq t < \iota_d\}$$

$$\mathcal{T}_{cdt_1} = \{t = t_1 + kn^c_{\max} : \ k \in I\!N \cup \{0\}, \ \iota_d \leq t < \tau_{d+1}\}$$

Equalities

$$\sum_{r\in R} \sum_{t\in T_{cdt_1}} x_{crt} + \sum_{r\in R} \sum_{t\in \mathcal{T}_{cdt_2}} x_{crt} = u_{cd}, \qquad \begin{array}{l} c \in C, \ d \in D, \\ \tau_d \leq t_1 < \tau_d + n^c_{\max}, \\ \iota_d \leq t_2 < \iota_d + n^c_{\max}, \end{array} \tag{26}$$

are valid for P_{UCTP}.

Proof.

$$\sum_{r\in R} \sum_{t\in T_{cdt_1}} x_{crt} + \sum_{r\in R} \sum_{t\in \mathcal{T}_{cdt_2}} x_{crt} \leq u_{cd}$$

follows from (7) and (21) when $c = c_1 = c_2$. To prove the opposite inequality, let us consider two cases. When $u_{cd} = 0$, is obvious. When $u_{cd} = 1$, it means that course c takes place in day d at least once during the consequent $n^c_{\min}$ hours in the morning or evening session. It means that

$$\sum_{r\in R} \sum_{t\in T_{cdt_1}} x_{crt} + \sum_{r\in R} \sum_{t\in \mathcal{T}_{cdt_2}} x_{crt} \geq 1$$

◇

The two following families of inequalities are derived from compactness inequalities (8) and (12) by sequential lifing.

Proposition 5.4 *Given intersection graph $G(V, E)$, let $H_{c_1 t_1 t_3}$ be a complete subgraph of G with property that every $x_{crt} \in H_{c_1 t_1 t_3}$ is adjacent in G to nodes $x_{c_1 r t_1}$ and $x_{c_1 r t_3}$ for all $r \in R$. Lifted Course Compactness inequalities*

$$\begin{array}{l} \sum\limits_{r \in R} (x_{c_1 r t_1} - x_{c_1 r t_2} + x_{c_1 r t_3}) + \sum\limits_{x_{crt} \in H_{c_1 t_1 t_3}} x_{crt} \leq 1, \\ c_1 \in C,\ d \in D,\ \tau_d \leq t_1 < t_2 < t_3 < \tau_{d+1} \end{array} \tag{27}$$

are valid for P_{UCTP}.

Proposition 5.5 *Given intersection graph $G(V, E)$, let $H_{g t_1 t_3}$ be a complete subgraph of G with property that every $x_{crt} \in H_{g t_1 t_3}$ is adjacent in G to nodes $x_{c_1 r t_1}$ and $x_{c_1 r t_3}$ for all $c_1 \in C_g$, $r \in R$. Lifted Class Compactness inequalities*

$$\begin{array}{l} \sum\limits_{c_1 \in C_g} \sum\limits_{r \in R} (x_{c_1 r t_1} - x_{c_1 r t_2} + x_{c_1 r t_3}) + \sum\limits_{x_{crt} \in H_{g t_1 t_3}} x_{crt} \leq 1, \\ g \in G,\ d \in D,\ \tau_d \leq t_1 < t_2 < t_3 < \tau_{d+1} \end{array} \tag{28}$$

are valid for P_{UCTP}.

5.1 Separation algorithms

Inequalities (22)–(25) can be easily checked by enumeration. Separation of inequalities (27) is implemented in the following maner.

Let $\bar{x}$ be the current fractional solution. For every class c_1 and day d we find t_1, t_2, t_3 by a greedy way in order to maximize

$$\sum_{r \in R} (x_{crt_1} - x_{crt_2} + x_{crt_3})$$

We find a set variables, which are adjacent to all $x_{c_1 r t_1}$ and $x_{c_2 r t_3}$ for all $r \in R$. Then, among of them, a clique with maximal weight (i.e. $H_{c_1 t_1 t_3}$) is found by MIP solver. The corresponding cut is added in case of violation.

A similar separation procedure is used for cuts (28).

6 Computational results

In this section we report on a computational experience with real-world instances utilizing a sourse more than 1000 students attending courses (kindly provided by the Facoltá di Ingegneria of Universitá del Sannio, Benevento, Italy). The test-bed consists of four instances: academic years 2001-2002 and 2002-2003, I and II semesters.

Below, we describe main features of the didatical organization. The week consists of five working days (from Monday to Friday). Each day is organized in two sessions, namely a morning session (9am-2pm) and an afternoon session (2pm-7pm). More details are reported in table 1, where *Name* is name of the instance , $\bar{c}$ is number of courses, $\bar{h}$ is total number of hours to be scheduled, i.e. $\bar{h} = \sum\limits_{c \in C} n_c$, $\bar{r}$ is number of available rooms,

$Name$	$\bar{c}$	h	$\bar{r}$	$\bar{g}$	$\bar{s}$	$\sharp Nodes$	$\sharp Arcs$
$Prob1$	64	233	12	14	48	4815	115025
$Prob2$	69	229	15	13	59	7595	332831
$Prob3$	57	212	9	11	43	4205	83745
$Prob4$	61	207	12	11	41	6670	223816

Table 1: Instance details

$\bar{g}$ is number of classes, $\bar{s}$ is number of teachers. $\sharp Nodes$ and $\sharp Arcs$ are, respectively, number of nodes and arcs in Set Packing relaxation.

In our tests, objective function penalty p_{ct} has been set to 1 if time period t has been undesirable for teacher of course c, 0 otherwise. With these penalties, optimal objective value returns minimum number of "undesired" time periods.

Computational experiments have been carried out on a Compaq EVO $W4000$ Personal Computer with Pentium IV-1.8 Ghz processor and 1 Gb RAM. We have used MIP solver ILOG Cplex 8.0 callable library [16] as Branch-and-Cut framework.

For the sake of clarity, we recall that, based on the propositions of sections 4 and 5, tightened formulation is defined by inequalities (2), (3), (5)–(8), (10), (12), (13)–(28).

We classify these inequalities into three groups according to the separation strategies we adopted:

1. Initial formulation contains constraints (2), (3), (5)–(7), (10), (14), (18), (25).

2. Because of their huge number, constraints (8), (12), (19)–(24), (26) are added dynamically, i.e. they are stored into a cut pool and added to the current problem as soon as they become violated during optimization. The pool is checked at every node of $B\&C$ tree.

3. Separation algorithms for the Clique, Lifted Odd-hole inequalities and inequalities (27), (28) are launched only at root node of $B\&C$ tree.

Computational results are reported in table 2. Column $Name$ shows the instance name, LB_{LP} is the lower bound of the LP-relaxation of the basic formulation (2)-(14), LB_{Cut} is the lower bound yielded after addition of cutting planes at root node of the $B\&C$ tree, BLB is best lower bound produced by the $B\&C$ algorithm, BUB is best upper bound produced by Branch-and-Cut algorithm ("—" means that no feasible solutions were found), $Guar$ is the optimality guarantee, computed as $\frac{BLB}{BUB} \cdot 100\%$ ("opt" means that optimal solution was found). $\sharp Nodes_{B\&C}$ and $Time$ are, respectively, number of Branch-and-Cut nodes and total computation time (CPU seconds). Computation time was limited to 2 hours. To demonstraite effectiveness of cutting planes introduced, we ran Branch-and-Cut algorithm with three different cut generation strategies:

Strategy 1. We use only constraints of the basic formulation (2)-(14). In this case, we cannot even find any feasible solution in two hours of computation time. Some-

$Name$	LB_{LP}	LB_{Cut}	BLB	UB	$Guar$	$\sharp Nodes_{B\&C}$	$Time$ (sec)
Strategy 1							
$Prob1$	9.40	11.60	21.40	—	—	4972	7200
$Prob2$	6.00	6.00	18.00	—	—	8628	7200
$Prob3$	11.00	13.00	22.00	—	—	27972	7200
$Prob4$	25.00	25.00	32.00	—	—	4206	7200
Strategy 2							
$Prob1$	9.40	24.50	33.00	33.00	opt	999	7038
$Prob2$	6.00	27.50	27.50	33.00	83.33%	410	7200
$Prob3$	11.00	22.99	28.00	28.00	opt	4844	4940
$Prob4$	25.00	41.47	43.00	44.00	97.73%	347	7200
Strategy 3							
$Prob1$	9.40	31.00	33.00	33.00	opt	32	348
$Prob2$	6.00	32.33	33.00	33.00	opt	13	681
$Prob3$	11.00	24.83	28.00	28.00	opt	591	893
$Prob4$	25.00	43.33	44.00	44.00	opt	5	512

Table 2: Computational results

times, values of LB_{LP} and LB are different, because of embedded cutting planes of MIP solver.

Strategy 2. We use cutting planes (18)-(24). With this strategy, $Prob1$ is solved to optimality in time very close to the limit of 2 hours. For $Prob2$ the upper bound coincides with optimal, but it is proved is proved only with 83.33% optimaly guarantee, $Prob3$ is solved to optimality in one hour and 25 minutes. For $Prob4$, a solution is found with 97.73% optimality guarantee.

Strategy 3. We use all the cutting planes families introduced in this paper. Clique inequalities, Odd-Hole inequalities and Lifted Compactness inequalities (27), (28) are identified only at the root node. This strategy yields optimal solution of all test instances in less than 15 minutes of CPU Time.

Acknowledgments

The authors wish to thank Antonio Sassano for the helpful comments.

References

[1] Abramson D., A very high speed architecture for simulated annealing, *IEEE Computer* 25 (1992) 27-36.

[2] Asratian A.S., de Werra D. A generalized class-teacher model for some timetabling problems, *European journal of operational research* 143 (2002) 531-542.

[3] Birbas T., Daskalaki S., Housos E. Timetabling for Greek High School, *Journal of the Operational Research Society* 48 (1997) 1191-1200.

[4] Borndorfer R., Weismantel R. Set Packing Relaxations of Some Integer Programs, *Mathematical Programming* 88 (2000) 425-450.

[5] Burke E., Carter M. (editors), The Practice and Theory of Automated Timetabling II: Selected Papers from the 2nd International Conference on the Practice and Theory of Automated Timetabling, University of Toronto, August 20th-22nd 1997, Springer Lecture Notes in Computer Science Series 1998, Volum 1408.

[6] Burke E., Erben W. (editors) The Practice and Theory of Automated Timetabling III: Selected Papers the 3rd International Conference on the Practice and Theory of Automated Timetabling, Konstanz, Germany, August 16th-18th 2000, Springer Lecture Notes in Computer Science Series 2001, Volume 2079.

[7] Burke E.K., Newall J.P., Weare R.F. A memetic algorithm for university exam timetabling, *The Practice and Theory of Automated Timetabling* **1153**, E.K. Burke and P. Ross editors (1996) 241–250.

[8] Burke E.K., Petrovic S. Recent Research Trends in Automated Timetabling, *European Journal of Operational Research* 140(2) (2002) 266-280.

[9] Burke E., Ross P. (editors), The Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference on the Practice and Theory of Automated Timetabling, Edinburgh August/September 1995, Lecture Notes in Computer Science 1996, Volulem 1153.

[10] Carter M.W. A survey of practical applications of examination timetabling algorithm, *Operations Research* 34(2) (1986) 193-202.

[11] Colorni A., Dorigo M., Maniezzo V. Mateheuristics for high school timetabling, *Computational Optimization and Application* 9 (1998) 275-298.

[12] Corne D., Ross P., Fang H.-L. Evolutionary timetabling: practice, prospects and work in progress, *In UK planning and Scheduling SIG Workshop* (1994)

[13] Ferland J.A., Roy S. Timetabling problem for university as assignment of activity to resources, *Computers and Operational Research* 12(2) (1985) 207-218.

[14] Gotlieb C.C. The construction of class-teacher timetables, in Proceeding of IFIP Congress (1962) 73-77.

[15] Hoffman K.L., Padberg M. Solving airline crew scheduling problems by branch-and-cut, *Management Science* 39(6) (1993) 657-682.

[16] ILOG CPLEX Reference manual, *ILOG* (2002).

[17] Grotschel M., Lovasz L., Schrijver A. Geometric algorithms and combinatorial optimization, *Springer-Verlag* (1993).

[18] Mannino C., Sassano A. An exact algorithm for the maximum stable set problem, *Computational Optimization and Applications* 3(4) (1994) 243-258.

[19] Schaerf A., Di Gaspero L. Local Search Techniques for Educational Timetabling Problems, *in Proceeding of the 6th International Symposium on Operational Research in Slovenia (SOR-'01)*, Preddvor, Slovenia (2001) 13-23

[20] Schaerf A. A survey of automated timetabling, *Artificial Intelligence Review* 13 (1999) 87-127.

[21] Tripathy A. School timetabling – a case in large binary integer linear programming, *Management Sciense* 30(12) (1984) 1473-1489.

[22] Waterer H., Johnson E.L., Nobili P., Savelsbergh M.W.P. The relation of time indexed formulations of single machine scheduling problems to the node packing problem, *Mathematical Programming* 93 (2002) 477-494.

[23] de Werra D. An introducing to timetabling, *European Journal of Operational Research* 19 (1985) 151-162.

[24] de Werra D., Asratian A.S., Durand S. Complexity of some types of timetabling problems, *Journal of Scheduling* 5 (2002) 171-183.

A Clique separation

Let $G(V, E)$ is a simple graph. Let us consider a polytope of the Set Packing problem

$$P_{SSP} = \left\{ y \in I\!B^{|V|} : \ y_i + y_j \leq 1, \ \forall ij \in E \right\}.$$

It is well-known that the inequality

$$\sum_{i \in K} y_k \leq 1 \tag{29}$$

defines a facet of P_{SST} if and only if $K \subset V$ is a *Maximum Clique*. The separation problem consists of finding maximum cliques for graph $G(V, E)$ for which corresponding inequalities (29) are violated with respect to given fractional solution $\bar{y}$. Our separation is very similar to procedure proposed by Hoffman and Padberg [15], but it is exact, i.e. we identify violated Clique inequalities if they exist. This separation is based on the fact that small problems can be solved quickly.

In the first stage of separation, we identify the most violated Clique inequalities on subgraph $\bar{G}(\bar{V}, E(\bar{V}))$. Here we denote with $E(W) = \{ij \in E : \ i \in W, \ j \in W\}$, where node set corresponds to fractional values of $\bar{y}$, i.e. $\bar{V} = \{i \in V : \ 0 < \bar{y}_i < 1\}$. We choose a node i from $\bar{V}$ with minimum degree. Let $star(i) = \{j : \ ij \in E\}$ (a subset of neighbors of i). Every clique, containing i, belongs to $star(i) \cup \{i\}$. Therefore, the most violated Clique inequality can be found by solving the Maximum Weighted Stable Set problem on the complement graph with nodes $star(i) \cup \{i\}$. If $|star(i) \cup \{i\}| \leq 16$, the problem is solved by enumeration, otherwise by MIP solver. If the problem value is more than 1, violated Clique inequality is found and can be enlarged with nodes of entire graph $G(V, E)$ in the second stage of separation. The first stage is repeated after deleting i from $\bar{V}$ until $\bar{V}$ is exhausted.

In the second stage of separation, violated Clique inequalities are enlarged to be facets, i.e. corresponding cliques are enlarged to be maximum cliques. Suppose, we have clique K, our purpose is to find the maximal clique containing K. It is done by the similar method as used in the first stage. We form the node set $M = \{j \in V : \ ij \in E \ \forall i \in K\}$ and solve the Maximal Clique problem over the nodes $K \cup M$ with fixed nodes from K in the solution. Computational experiments show that this problem can be solved faster through the Stable Set problem on the complement graph, because of sparsity of the complement graph. Again, this problem is solved either by enumeration, if the number of nodes is less or equal 16, or by MIP solver.

B Odd-Hole separation

An *odd-hole* in $G(V, E)$ is a set of nodes $W = \{w_1, w_2, ..., w_{2k+1}\}$ for $k \geq 2$ such that node u_{i-1} is joined by an edge to node w_i for $i = 1, ..., 2k + 1$, where $w_0 = w_{2k+1}$ and no other pair of nodes of W is joined by an edge. The inequality

$$\sum_{w \in W} y_w \leq \frac{|W| - 1}{2} \tag{30}$$

is called an *odd-hole inequality*. It is valid for P_{SSP} [15].

Odd-Hole inequalities can be separated in polynomial time [17]. Here, because of the large size of the instances, we apply the fast heuristic proposed by Hoffman and Padberg [15].

Let us consider graph $\bar{G}(\bar{V}, E(\bar{V}))$. Choose a node $w \in \bar{V}$ and build a "layered" graph rooted in it. On the first layer, there are all neighbors of w, on the second layer, there are remaining neighbors of the first layer nodes and so on. So, the shortest path from level k to root w contains exactly k edges. Assign edgeweights of $1 - \bar{y}_i - \bar{y}_j$ for all edges $ij \in E$ those are in the layered graph. Pick two adjacent nodes i and j on some layer k, find the shortest path from the root to i and set the edgeweights of this path to a big value L, say. Then, find the shortest path from the root to node j. If the length of the second path is less than L, we have an odd-hole formed by the two paths and can check if it is violated or not.