

Strong Bounds with Cut and Column Generation for Class-Teacher Timetabling

Haroldo Gambini Santos · Eduardo Uchoa ·
Luiz Satoru Ochi · Nelson Maculan

Received: date / Accepted: date

Abstract This work presents an integer programming formulation for an important variant of the Class-Teacher Timetabling problem, which considers the satisfaction of teacher preferences and also the proper distribution of lessons throughout the week. The formulation contains a very large number of variables and is enhanced by cuts. Therefore, a cut and column generation algorithm to solve its linear relaxation is provided. The lower bounds obtained are very good, allowing us to prove the optimality of previously known solutions in three formerly open instances.

Keywords Integer Programming · Timetabling · Column Generation · Cut Generation

1 Introduction

The Class-Teacher Timetabling Problem (CTTP) concerns scheduling a set of encounters between teachers and classes over a set of periods, typically covering five week days, satisfying a set of constraints. The *basic* constraints that must be satisfied are: (i) avoid conflicts in each period: no teacher should be allocated to more than one class and no class should receive more than one teacher; (ii) consider the periods of unavailability of each teacher; and (iii) allocate the right number of periods for each teacher-class pair. This classical version of the CTTP [18] was shown to be NP-Complete [17].

The manual construction of timetables in educational institutions is often a laborious and time-consuming task. Besides the basic constraints, a good timetable should consider many other requirements, such as institutional, pedagogical and personal (staff related) needs. In fact, the criteria that define the quality and even the feasibility of a timetable often depends on the specific educational system. However, some common assumptions are made in most works in the literature. For example, timetable compactness: teachers prefer having all their classes concentrated in as few days of the week as possible. Even in a single day, “holes”, periods of inactivity between two classes, should be avoided. The compactness requirements are particularly important in countries (like Brazil) where teachers may work in more than one institution. Those requirements were considered in [11, 26, 3, 10, 9, 14, 20, 4, 13, 23, 15, 28]. In most of these works, compactness

is treated as a *soft constraint* [16], i.e., this is actually considered in the objective function as a measure to evaluate timetables. The solution space definition, specified by the *hard constraints*, usually does not include many other constraints besides the above mentioned *basic* constraints. Another very common type of soft constraint in school timetabling is related to the length of the lessons: for some teacher-class pairs it is interesting to have at least a certain number of double lessons, where the teacher lectures to the same class for two consecutive periods [26,11,28,27]. On the other hand, one should avoid more than one lesson (excepting double lessons) to a class by the same teacher in a single day. This paper proposes a new formulation that considers these timetabling goals and constraints. It is formally described in Section 2 and we name it the Class-Teacher Timetabling Problem with Compactness Constraints (CTTPC).

Since exact methods, which try to find proven optimal solutions, can demand unrealistic amounts of processing time on most timetabling variants, the development of heuristics which are effective in practice has received much attention from researchers. In this sense, sophisticated metaheuristics have been developed to try to tackle real world problems. These methods include Simulated Annealing [1], Evolutionary Algorithms [14,10,2], Tabu Search [11,26,10,28] and some of their hybrids. No theoretical performance guarantees are available for those heuristics and their evaluation strongly depends on empirical results. In many works, the quality of the proposed timetables is measured by their acceptance by the educational institution, which is very subjective. Ideally, every new heuristic should be compared with previously proposed methods over an established set of representative benchmark instances. Unfortunately, this is not yet the case in CTTPCs. In fact, each group of authors considers different variants of the problem (sometimes only slightly different), making such comparisons difficult.

There is another way of evaluating the quality of a timetabling heuristic: comparing the solutions costs with dual bounds. Suppose that one seeks timetables minimizing weighted penalties for violating soft constraints. In this minimization case, dual bounds are valid lower limits on the cost of an optimal solution. Dual bounds can be often obtained from solving the linear programs obtained by removing the integrality constraints of a Mixed Integer Programming (MIP) formulation of the timetabling problem. However, the quality of these limits varies significantly depending on the formulation used. Compact formulations (those without too many variables and constraints) of timetabling problems usually yield poor dual bounds.

Two of the most used MIP techniques for improving dual bounds on hard combinatorial optimization problems are cut and column generation. This work explores both techniques for producing high quality dual bounds for CTTPC instances. A novel feature in this kind of algorithm is introduced: the use of Fenchel cuts over knapsack-like constraints induced by the columns.

The paper is organized as follows: Section 2 formally introduces the problem by presenting a compact formulation, Section 3 introduces the extended formulation, detailing the cut and column generation procedures. In Section 4 computational experiments with the different formulations are presented. Finally, Section 5, contains the conclusions.

2 A Compact Formulation

The compact formulation for the CTTPC will be helpful to formally describe the problem and for comparisons with the extended formulation presented later. Consider first the following problem data:

- D : set of week days;
- P : set of periods by day. It is assumed that all days have the same number of periods, numbered from 1 to $|P|$;
- T : set of teachers;
- $\tilde{T}_{|T| \times |D| \times |P|}$: teachers' availability matrix, where $\tilde{t}_{tdp} = 1$ indicates that teacher t is available for teaching at day d and period p , $\tilde{t}_{tdp} = 0$ otherwise;
- C : set of classes;
- $\tilde{R}_{|T| \times |C|}$: requirements matrix, where \tilde{r}_{tc} indicates how many lessons teacher t must teach to class c ;
- $\tilde{M}_{|T| \times |C|}$: daily limits matrix, where \tilde{m}_{tc} indicates the maximum number of lessons that teacher t can teach to class c in one day;
- $\tilde{G}_{|T| \times |D|}$: inferred from \tilde{T} , subset $\tilde{G}_{td} \subset P$ of periods in teacher's t agenda at day d which can be the start of a double lesson;
- $\tilde{G}'_{|T| \times |C|}$: double lesson requests matrix, where \tilde{g}'_{tc} indicates how many double lessons are requested by teacher t for lessons with class c , must be compatible with \tilde{M} ;
- $W'_{|T|}$: penalty vector, where w'_t indicates the cost of a work day in teacher's t agenda;
- $W''_{|T|}$: penalty vector, where w''_t indicates the cost of a hole in teacher's t agenda;
- $W'''_{|T|}$: penalty vector, where w'''_t indicates the cost of not satisfying a double lesson request of teacher t .

Additionally, consider the following decision variables:

$$x_{tcdp} = \begin{cases} 1 & \text{if teacher } t \text{ is teaching to class } c \text{ at day } d \text{ and period } p \\ 0 & \text{otherwise} \end{cases}$$

This way, the CTTPC can be formulated as a MIP, which will be denoted here by \mathcal{F}_1 :

$$\begin{aligned}
& \text{minimize} \\
& \sum_{t \in T} \sum_{d \in D} w'_t v_{td} + \sum_{t \in T} \sum_{d \in D} w''_t h_{td} + \sum_{t \in T} \sum_{c \in C} w'''_t g_{tc} \quad (1) \\
& \text{subject to} \\
& \sum_{d \in D} \sum_{p \in P} x_{tcdp} = \tilde{r}_{tc} \quad \forall t \in T, c \in C \quad (2) \\
& \sum_{t \in T} x_{tcdp} \leq 1 \quad \forall c \in C, d \in D, p \in P \quad (3) \\
& \sum_{c \in C} x_{tcdp} \leq \tilde{t}_{tdp} \quad \forall t \in T, d \in D, p \in P \quad (4) \\
& \sum_{p \in P} x_{tcdp} \leq \tilde{m}_{tc} \quad \forall t \in T, c \in C, d \in D \quad (5) \\
& v_{td} \geq \sum_{c \in C} x_{tcdp} \quad \forall t \in T, d \in D, p \in P \quad (6) \\
& \bar{a}_{td} \leq (|P| + 1) - (|P| + 1 - p) \sum_{c \in C} x_{tcdp} \quad \forall t \in T, d \in D, p \in P \quad (7) \\
& \underline{a}_{td} \geq p \sum_{c \in C} x_{tcdp} \quad \forall t \in T, d \in D, p \in P \quad (8) \\
& h_{td} \geq \underline{a}_{td} - \bar{a}_{td} + v_{td} - \sum_{c \in C} \sum_{p \in P} x_{tcdp} \quad \forall t \in T, d \in D \quad (9) \\
& y_{tcdp} \leq x_{tcdp} \quad \forall t \in T, c \in C, d \in D, p \in \hat{G}_{td} \quad (10) \\
& y_{tcdp} \leq x_{tcdp+1} \quad \forall t \in T, c \in C, d \in D, p \in \hat{G}_{td} \quad (11) \\
& g_{tc} \geq \tilde{g}_{tc} - \sum_{d \in D} \sum_{p \in \hat{G}_{tc}} y_{tcdp} \quad \forall t \in T, c \in C \quad (12) \\
& \bar{a}_{td} \geq 0 \quad \forall t \in T, d \in D \quad (13) \\
& \underline{a}_{td} \geq 0 \quad \forall t \in T, d \in D \quad (14) \\
& g_{tc} \geq 0 \quad \forall t \in T, c \in C \quad (15) \\
& h_{td} \geq 0 \quad \forall t \in T, d \in D \quad (16) \\
& y_{tcdp} \geq 0 \quad \forall t \in T, c \in C, d \in D, p \in \hat{G}_{td} \quad (17) \\
& v_{td} \geq 0 \quad \forall t \in T, d \in D \quad (18) \\
& x_{tcdp} \in \{0, 1\} \quad \forall t \in T, c \in C, d \in D, p \in P \quad (19)
\end{aligned}$$

In this formulation, Constraints (2) ensure the required number of meetings between teacher and classes throughout the week; (3) avoid conflicts in classes timetables; (4) impose respect for teachers' availability and forbid conflicts in teachers' timetables; (5) limit the number of lessons for each teacher and class pair per day; (6) link each variable v_{td} , which indicates if there is some activity for teacher t in a day d , to the decision variables x . The number of holes for a given teacher t at day d is measured by variables h_{td} , linked to variables \bar{a}_{td} and \underline{a}_{td} that indicate, respectively, the first and the last period with some class for that teacher in that day. All those variables are connected with the decision variables by Constraints (7), (8), and (9). The existence of a double lesson for teacher t and class c starting at day d and period p is indicated

by variables y_{ptdp} , present in Constraints (10) and (11). Finally, the non-satisfaction of the minimal number of double lessons throughout the week for teacher t and class c is measured by variables g_{tc} , in Constraints (12).

Although \mathcal{F}_1 is a weak formulation, a simple and effective cut can be added to improve its lower bound, considering the minimum number of working days for teachers, as proposed by [27]:

$$\sum_{d \in D} v_{td} \geq \max \left\{ \left\lceil \frac{\sum_{c \in C} \tilde{r}_{tc}}{|P|} \right\rceil, \max_{c \in C} \left\{ \left\lceil \frac{\tilde{r}_{tc}}{\tilde{m}_{tc}} \right\rceil \right\} \right\} \quad \forall t \in T. \quad (20)$$

The first term in the maximum can be possibly improved by considering that teachers may not be available for $|P|$ periods on all days.

3 The Extended Formulation

In this section an alternative formulation to the CTPP will be presented, which has a very large number of variables. This reformulation approach is often used to obtain linear relaxations giving better bounds and also less prone to symmetry problems [6]. We further strength this formulation by including an exponential number of constraints, thus it will require a cut and column generation algorithm for its solution.

The proposed formulation will be denoted by \mathcal{F}_2 . Consider, besides data presented in Section 2, that \tilde{T}_{td} is the set of all possible non-empty class allocations for teacher t in workday d . For every $j \in \tilde{T}_{td}$, define \check{x}_{tdjcp} as the binary constant which indicates if teacher t teaches in day d to class c in period p of allocation j ; the cost c_{tdj} is given by w'_t plus the number of holes in allocation j times w''_t ; and define \check{g}_{tdjc} as the number of double lessons for teacher t and class c included in allocation j . The decision variable λ_{tdj} indicates if allocation j will be selected for the work day d of teacher t . This way, \mathcal{F}_2 is:

$$\begin{aligned} & \text{minimize} \\ & \sum_{t \in T} \sum_{d \in D} \sum_{j \in \tilde{T}_{td}} c_{tdj} \lambda_{tdj} + \sum_{t \in T} \sum_{c \in C} w'''_t g_{tc} \end{aligned} \quad (21)$$

$$\begin{aligned} & \text{subject to} \\ & \sum_{j \in \tilde{T}_{td}} \lambda_{tdj} \leq 1 \quad \forall t \in T, d \in D \end{aligned} \quad (22)$$

$$\sum_{t \in T} \sum_{j \in \tilde{T}_{td}} \check{x}_{tdjcp} \lambda_{tdj} \leq 1 \quad \forall c \in C, d \in D, p \in P \quad (23)$$

$$\sum_{d \in D} \sum_{p \in P} \sum_{j \in \tilde{T}_{td}} \check{x}_{tdjcp} \lambda_{tdj} = \tilde{r}_{tc} \quad \forall t \in T, c \in C \quad (24)$$

$$\sum_{d \in D} \sum_{j \in \tilde{T}_{td}} \check{g}_{tdjc} \lambda_{tdj} + g_{tc} \geq \tilde{g}_{tc} \quad \forall t \in T, c \in C \quad (25)$$

$$g_{tc} \geq 0 \quad \forall t \in T, c \in C \quad (26)$$

$$\lambda_{tdj} \in \{0, 1\} \quad \forall t \in T, d \in D, j \in \tilde{T}_{td} \quad (27)$$

Constraints (22) ensure that at most one allocation is selected by teacher and day; (23) avoid conflicts in classes; (24) enforce the number of lessons for each teacher

and class. Constraints (25) measure, with auxiliary variables g_{tc} , the number of non-satisfied double lessons requests for a teacher t on class c . Although the large number of variables in \mathcal{F}_2 may prohibit its direct use, the exact solution of the linear relaxation of \mathcal{F}_2 , denoted here by \mathcal{F}_{2r} , can be done by column generation [12]. For this technique, one starts with a small subset of columns in the linear program and, at each iteration, new promising columns are identified by its reduced cost with respect to the current dual variables. These columns are progressively inserted in the linear program until it is proven that no column outside that linear program would improve its solution.

3.1 Column Generation

Initially, for solving \mathcal{F}_{2r} , no variables λ are present, but artificial variables (with high costs) are inserted, to allow a first feasible linear solution. The *pricing* problem consists in finding, for each teacher and day, the allocation with the smallest reduced cost. We denote here by μ, ν, π e κ the dual variables associated to Constraints (22), (23), (24) and (25), respectively. The pricing problem related to teacher t and day d , denoted by \mathcal{P}_{td} can be formulated as a following MIP. Let binary variables x_{cp} ($c \in C, p \in P$) indicate that teacher t has a lesson with class c at day d and period p :

$$\mathcal{P}_{td} = \left\{ \begin{array}{l} \text{minimize} \\ \quad \bar{c}_{td} = w'_t + w''_t h - \mu_{td} - \sum_{c \in C} \sum_{p \in P} [(\pi_{tc} + \nu_{cdp})x_{cp}] - \sum_{c \in C} \sum_{p \in \hat{G}_{td}} \kappa_{tc} y_{cp} \\ \text{subject to} \\ \quad \sum_{p \in P} x_{cp} \leq \min\{\bar{m}_{tc}, \bar{r}_{tc}\} \quad \forall c \in C \\ \quad \bar{a} \leq \sum_{c \in C} (p - |P|)x_{cp} + |P| \quad \forall p \in P \\ \quad \underline{a} \geq \sum_{c \in C} p x_{cp} \quad \forall p \in P \\ \quad h = 1 + \underline{a} - \bar{a} - \sum_{c \in C} \sum_{p \in P} x_{cp} \\ \quad \sum_{c \in C} x_{cp} \leq \bar{t}_{tdp} \quad \forall p \in P \\ \quad y_{cp} \leq x_{cp} \quad \forall c \in C, p \in \hat{G}_{td} \\ \quad y_{cp} \leq x_{c,p+1} \quad \forall c \in C, p \in \hat{G}_{td} \\ \quad h \geq 0 \\ \quad x_{cp} \in \{0, 1\} \quad \forall t \in T, p \in P \end{array} \right.$$

Problems \mathcal{P}_{td} , $\forall t \in T, d \in D$, are solved at each iteration, increasing the problem size with at most $(|T| \times |D|)$ columns per iteration. The optimal solution for \mathcal{F}_{2r} is attained when $\bar{c}_{td} \geq 0 \forall t \in T, d \in D$. Although no known algorithm can solve general instances of \mathcal{P}_{td} in polynomial time, one observation must be made about the hardness of solving the presented pricing problem in practice. The number of possible allocations grows exponentially with $|P|$. However, in real school timetabling problems the value of $|P|$ is very limited, typically 5 or 6 by shift, depending on the educational system. Larger instances may contain more teachers and classes, but $|P|$ remains constant. One can price the solutions of \mathcal{P}_{td} by enumeration in $O(|C|^{|P|})$ time, which is polynomial for a fixed $|P|$. In practice, this pricing by enumeration can be performed in reasonable time, even for large instances.

3.2 Cut Generation

Some cuts that are valid for \mathcal{F}_1 can be translated to \mathcal{F}_2 . For example, the right-hand side of (20) gives a minimum on the number of lambda variables that must be chosen for each teacher. However, \mathcal{F}_2 allows the natural definition of some new strong cuts.

The cuts that will be presented are related to the satisfaction of the total weekly workload considering the possible daily choices in the number of lessons for teachers and for meetings between teachers and classes. We will introduce new variables and constraints to \mathcal{F}_2 in order to make easier the expression of these cuts. This formulation, denoted here by \mathcal{F}_2' incorporates binary variables a_{tdl} which indicate if teacher t has a workload of exactly l lessons at day d , as well binary variables a_{tcdl} , which indicate if teacher t teaches exactly l lessons for class c at day d . Let \check{T}_{tdl} be the subset of columns of \check{T}_{td} such that the total number of lessons in every column from \check{T}_{tdl} is exactly l and \check{T}_{tcdl} the subset of \check{T}_{td} such that every column of \check{T}_{tcdl} has exactly l lessons of teacher t to class c , the new variables are related to the λ variables by:

$$\sum_{j \in \check{T}_{tdl}} \lambda_{tdj} = a_{tdl} \quad \forall t \in T, d \in D, l \in P \quad (28)$$

$$\sum_{j \in \check{T}_{tcdl}} \lambda_{tdj} = a_{tcdl} \quad \forall t \in T, c \in C, d \in D, l \in P \quad (29)$$

The pricing problem for teacher t and day d in formulation \mathcal{F}_2' , denoted here by \mathcal{P}'_{td} , must be augmented to reflect the inclusion of λ variables in two additional constraints. But this does not make much difference, as long as the pricing is being solved by enumeration.

We will now strengthen the formulation by introducing inequalities that are valid for two families of 0-1 polyhedra. Polyhedra χ_t is given by the 0-1 solutions of

$$\sum_{d \in D} \sum_{l \in \{1, \dots, \sum_{p=1}^{|P|} \check{t}_{tdp}\}} l a_{tdl} = \sum_{c \in C} \check{r}_{tc} \quad \forall t \in T. \quad (30)$$

Polyhedra χ_{td} is given by the 0-1 solutions of

$$\sum_{d \in D} \sum_{l \in \{1, \dots, \min\{\sum_{p=1}^{|P|} \check{t}_{tdp}, \check{m}_{tc}\}\}} l a_{tcdl} = \check{r}_{tc} \quad \forall t \in T, c \in C \quad (31)$$

One way to derive valid cuts from those polyhedra would be consider known facets for the knapsack constraints (30) or (31), like those provided in [5, 19, 29]. However, since the number of 0-1 solutions in those polyhedra is quite small, we opted for the more generic approach of Fenchel separation, as proposed by Boyd [7, 8]. This technique separates facets by solving a series of linear programs containing one constraint for each possible solution. This approach has shown to be very cheap and efficient way to obtain strong cuts. Again, this is possible due to the natural limitation in the number of periods $|P|$.

Algorithm 1 CCG**Input:** CTTTPC Instance.**Output:** Solution vector $\bar{\lambda}$ and dual bound.

```

1: initialize the linear program  $\mathcal{LP}$  using formulation  $\mathcal{F}'_2$ , considering only artificial variables and  $\lambda$  columns corresponding to inactivity days;
2: add into  $\mathcal{LP}$  cuts related to the minimum working days of teachers;
3:  $newColumns \leftarrow 1$ ;  $newCuts \leftarrow 0$ ;
4: while ( $newCuts + newColumns > 0$ ) do
5:   solve  $\mathcal{LP}$ , update solution ( $\bar{\lambda}$  and  $\bar{a}$ ) and dual variables;
6:    $newColumns \leftarrow 0$ ;
7:   for all  $t \in T$  do
8:     for all  $d \in D \mid \sum_{p \in P} \tilde{t}_{tdp} > 0$  do
9:       solve  $\mathcal{P}'_{td}$ ;
10:      if  $\bar{c}_{td} < 0$  then
11:        add new column into  $\mathcal{LP}$ ;
12:         $newColumns \leftarrow newColumns + 1$ ;
13:      end if
14:    end for
15:  end for
16:   $newCuts \leftarrow 0$ ;
17:  if  $newColumns = 0$  and  $fractionalSolution(\bar{a})$  then
18:    for all  $t \in T$  do
19:      solve  $\mathcal{S}_t$ ;
20:      if Violated cut found then
21:        Add into  $\mathcal{LP}$ ;
22:         $newCuts \leftarrow newCuts + 1$ ;
23:      end if
24:      for all  $c \in C$  do
25:        solve  $\mathcal{S}_{tc}$ ;
26:        if Violated cut found then
27:          Add cut into  $\mathcal{LP}$ ;
28:           $newCuts \leftarrow newCuts + 1$ ;
29:        end if
30:      end for
31:    end for
32:  end if
33: end while
end.

```

3.3 The Column and Cut Generation Algorithm

The algorithm for columns and cuts generation, denoted here by CCG, solves to optimality the linear relaxation of \mathcal{F}'_2 complemented with the addition of cuts. The algorithm performs, at each iteration (lines 4-33), the addition of columns or cuts. The addition of cuts starts when no more columns with negative reduced cost are left out of the linear program \mathcal{LP} , indicating that the optimal solution of \mathcal{LP} was attained. The insertion of new cuts requires the solution of new pricing problems. This process continues until no more cuts are inserted.

4 Computational Experiments

Computational experiments were made in order to evaluate the bounds provided by different formulations. The column and cut generation code was implemented in C++, linear programs were solved by CPLEX 10[21]. The reported runs were over a Dell

inst.	$ T $	$ C $	$\sum_{t \in T} \sum_{c \in C} \tilde{r}_{tc}$	$\sum_{t \in T} \sum_{c \in C} \tilde{g}_{tc}$	sr	$\sum_{t \in T} \sum_{d \in D} \tilde{t}_{td} $
1	8	3	75	21	0.43	13,540
2	14	6	150	29	0.50	419,270
3	16	8	200	4	0.30	221,420
4	23	12	300	66	0.18	2,433,421
5	31	13	325	71	0.58	2,297,130
6	30	14	350	63	0.52	2,188,405
7	33	20	500	84	0.39	4,288,890

Table 1 Instance characteristics.

Optiplex G620 computer with a Pentium D 3.0 GHz processor and 2GB of RAM, the operating system was Linux.

The instance set (Table 1), is the same used in [28, 27, 24]. Instances 1 and 3 are from [22], the first is an artificial instance and 3 is from a high school in the city of Brasília. The remaining instances were collected by Souza [27] in high schools from the State of Minas Gerais, excepting 7, which is artificially built by considering an augmentation of other instances. All instances correspond to the timetabling problem of one week, where $|D| = |P| = 5$. Columns 2 to 5 in Table 1 indicate the number of teachers, classes, class-teacher encounters and double lesson requests for each instance. The column sparseness ratio is computed by considering the total number of lessons and the total number of unavailable periods as follows:

$$sr = \frac{|T| \times |D| \times |P| - (\sum_{t \in T} \sum_{c \in C} \tilde{r}_{tc} + \sum_{t \in T} \sum_{d \in D} \sum_{p \in P} (1 - \tilde{t}_{tdp}))}{|T| \times |D| \times |P|}$$

Lower sparseness ratios indicate more restrictive problems. The last column gives the number of possible allocations of teachers to classes in all workdays, each such allocation corresponds to a λ variable in Formulation \mathcal{F}_2 . In all instances the cost of a teacher workday is set as 9, the cost of each hole inside the workday of a teacher is 3 and each request for a double lesson that is unmet costs 1.

Table 2 shows results of different formulations. Column UB gives the best known upper bound for each instance. Those bounds were produced by very long runs of the heuristics described in [24] and [25]. Columns LB show the lower bounds obtained by each formulation, columns gap are the percent distance from the upper bound and columns $time$ are the running times in seconds. \mathcal{W} denotes the set of cuts related to the minimum number of working days for teachers (20), while *Fenchel* are the cuts described in Section 3.2. This last formulation is solved using the CCG algorithm. Formulations \mathcal{F}_2 and $\mathcal{F}_2 \cup \mathcal{W}$ are solved with a lighter version of CCG, without the Fenchel cut generation phase. Best results are shown in bold. It can be seen that bounds obtained by the full CCG algorithm are very strong. For the first three instances the lower bound matched the best known solution, proving its optimality. The running times were quite reasonable, except perhaps in instance 7. In this case we observed that the significant increase in cpu time, with respect to solving $\mathcal{F}_2 \cup \mathcal{W}$, was not caused by the Fenchel cut separation routine. In fact, cut separation was always fast in our experiments, but the need for additional calls of the column generation phase, implying many re-optimizations of the linear program, incurred considerable additional computing time.

For the sake of illustration, we present in Figure 1 the best known solution for instance 4. The cost of this solution was shown by CCG to be at most one unit away

Inst.	f_1			$f_1 \cup W$			f_2			$f_2 \cup W$			$f_2 \cup W \cup Fenchel$		UB	
	LB	gap	time	LB	gap	time	LB	gap	time	LB	gap	time	gap	time		
1	135	49.6	0.2	189.0	6:9	0.2	189.0	6:9	0.2	196.8	2.7	0.3	202.0	0.0	0.5	202
2	270	23.3	1.2	333.0	0.0	1.9	333.0	0.0	1.9	333.0	0.0	10.2	333.0	0.0	11.4	333
3	360	17.5	0.5	414.0	2:2	1.0	414.0	2:2	1.2	419.3	0.9	8.8	423.0	0.0	4.8	423
4	600	8.8	3.1	639.0	2:2	4.2	639.0	2:2	4.2	643.0	1.6	110.6	652.0	0.2	91.2	653
5	585	30.9	6.4	756.0	1.3	13.4	756.0	1.3	11.9	756.0	1.3	93.9	762.0	0.5	179.4	766
6	630	20.6	8.0	738.0	3.0	17.6	738.0	3.0	16.8	738.0	3.0	119.8	756.0	0.5	231.1	760
7	900	14.3	23.7	999.0	3.0	41.1	999.0	3.0	43.3	999.0	3.0	321.5	1,017.0	1.2	2,678.3	1,029

Table 2 Results for different formulations.

Teacher\Period	Day 1				Day 2				Day 3				Day 4				Day 5				
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	X	X	X	X	6	12	3	8					1	6		8	1	1	2	7	
2	10	11	11	9	12	12	3	5	11	11	5	12	2	10	10	5	5	10	3	9	
3	X	X	X	X	4				X	X	X	X	4			X	X	X	X	X	
4				1	4															X	
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
6	X	X	X	X	9	5	11	12	6		12	1	7	11	7	8	10	5	8	7	
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	3	2	2	4	
8	X	X	3	2	8	3	4	8	1	10	12	9	10	4	11	9	2	1	12	11	
9	6	5	6	5	7															7	
10	1	4	5	3	1				4	2	2	3	5						X	X	
11	8	6	7	12	11				6	12	9	9	10	10	8	7	11		X	X	
12	3	1	1	6	6	7	4	4	2	5	5	1	6	6	3	2	2	4	7	5	
13	X	X	X	X	8	8	9	11	11	10	10	8	9	12	12	11	9	12	12	11	
14	2	12	4	11	12				1	1	3	2	3			4	3	1	11	4	
15	7	10	9	8	5	5	9	10	10	7			6	9	5	6	7		6	8	
16	5	2	X	X	X	2	1	2	6	3	3	4	4	5	1	3	X	X	6	5	
17	9	8	12	7	10				9	7	11	10	8	8	11	12	7				
18	4	7	2	4	3	1	3	5	7	1	2	3	6	5	1	2	4	6	X	X	
19	X	X	X	X	X	X	X	X	X	8	8	X	X	X	X	X	X	X	X	X	
20	X	X	X	X	X	X	11	12	9	12	X	X	X	X	X	X	X	9	10	10	
21	11	9	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	12	X	
22	X	X	X	X	X	10	1	2	4		X	X	X	X	X	X	X	X	3	9	
23	12	3	8	10	2	10	2	6	5	9	7	6	11	4	7	5	9	1	4	8	
																				11	1
																				3	9
																				12	11
																				1	3
																				12	11
																				3	9
																				12	11
																				1	3
																				12	11

Fig. 1 Suboptimal solution - instance 4 - cost: 653 - lower bound by CCG: 652

from the optimal. One characteristic which can be observed in this solution is the occurrence of non-contiguous lessons to classes in some days for some teachers (e.g.: teacher 4, day 4, lessons to class 1). We could easily insert a new constraint to avoid allocations of this type, which are not desirable in general. This new constraint would restrict the search space and, probably, speed-up the CCG algorithm. However, with this new formulation we would not be able to compare the lower bounds with the aforementioned results of already developed heuristics.

5 Conclusions

In this work we proposed a new MIP formulation with a large number of constraints and columns for the CTTPC. Computational experiments with an algorithm to solve the linear relaxation of this formulation have shown that this formulation can provide strong lower bounds in reasonable times. This allowed us, for the first time, to determine the optimal solutions of some CTTPC instances that were available since the eighties and provided very tight limits for other instances. It is important to observe that the cut and column generation approach presented here can be possibly adapted to other variants of the class-teacher timetabling problem. For example, the proposed Fenchel cuts are based on structures that are very common in other timetabling problems.

References

1. D. Abramson. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, 37:98–113, 1991.
2. D. Abramson and J. Abela. A parallel genetic algorithm for solving the school timetabling problem. In *Proceedings of the 15th Australian Computer Science Conference*, 1992.
3. R. Alvarez-Valdés, G. Martin, and M. Tamarit. Constructing good solutions for the spanish school timetabling problem. *Journal of the Operational Research Society*, 47:1203–1215, 1996.
4. P. Avella and I. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8:497–514, 2005.
5. E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 1975.
6. C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.
7. E.A. Boyd. Fenchel cutting planes for integer programming. *Operations Research*, 42:53–64, 1992.
8. E.A. Boyd. Solving 0/1 integer programs with enumeration cutting planes. *Annals of Operations Research*, 50:61–72, 1994.
9. J. Caldeira and C. Agostinho. School timetabling using genetic search. In *Practice and Theory of Automated Timetabling*, Toronto, 1997.
10. A. Colomi, M. Dorigo, and V. Maniezzo. Metaheuristics for high-school timetabling. *Computational Optimization and Applications*, 9(3):277–298, 1998.
11. D. Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76:98–110, 1994.
12. G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 1960.
13. S. Daskalaki. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153:117–135, 2004.
14. C. Di Stefano and A. Tettamanzi. An evolutionary algorithm for solving the school timetabling problem. In E. Boers et al., editor, *Applications of Evolutionary Computing. EvoWorkshops 2001*, pages 452–462, 2001.
15. A. Drexler and F. Salewski. Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, 102(1):193–214, October 1997.

-
16. H. A. Eiselt and G. Laporte. Combinatorial optimization problems with soft and hard requirements. *Journal of the Operational Research Society*, 38(9):785–795, 1987.
 17. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal of Computing*, 5(4):691–703, December 1976.
 18. C.C. Gotlieb. The construction of class-teacher timetables. In *Proceeding IFIP congress 1962*, pages 73–77, Amnsterdam, 1963.
 19. P.L. Hammer, E.L. Johnson, and U.N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8:179–206, 1975.
 20. A. Hertz. Tabu search for large scale timetabling problems. *European Journal or Operational Research*, 54:39–47, 1991.
 21. ILOG S.A. *ILOG CPLEX 10.0 User's Manual*, 2006.
 22. S.S. Mata. O problema do horário na escola do segundo grau: Modelagem e implementação. Master's dissertation, Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ, 1989.
 23. K. Papoutsis, C. Valouxis, and E. Housos. A column generation approach for the timetabling problem of greek high schools. *Journal of the Operational Research Society*, 54:230–238, 2003.
 24. H. G. Santos, L.S. Ochi, and M.J.F. Souza. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *Journal of Experimental Algorithmics (JEA)*, 10, 2005.
 25. Haroldo G. Santos, Luiz Satoru Ochi, and E. B. Uchoa. Combining metaheuristics and integer programming on school timetabling problems (*abstract*). In *Proceedings of the 19th International Symposium on Mathematical Programming*, 2006.
 26. A. Schaerf. Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics Part A:systems and Humans*, 29(4):368–377, July 1999.
 27. M.J.F. Souza. *Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas*. PhD thesis, Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
 28. M.J.F. Souza, L.S. Ochi, and N. Maculan. *Metaheuristics: Computer Decision-Making*, chapter A GRASP-Tabu Search Algorithm for solving School Timetabling Problems, pages 659–672. Kluwer Academic Publishers, 2004.
 29. L.A. Wolsey. Faces for a linear inequality in 0-1 variables. *Mathematical Programming*, 1975.