THE IMPACT OF COMPACTNESS REQUIREMENTS ON THE RESOLUTION OF HIGH SCHOOL TIMETABLING PROBLEM

Árton P. Dorneles
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brazil
apdorneles@inf.ufrgs.br

Olinto C. B. de Araújo
Colégio Técnico Industrial de Santa Maria - Universidade Federal de Santa Maria - UFSM
Santa Maria - Brazil
olinto@ctism.ufsm.br

Luciana S. Buriol
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brazil
buriol@inf.ufrgs.br

ABSTRACT

The school timetabling is a classic optimization problem that takes a large number of variables and constraints into account. Due to the combinatorial nature of this problem, it is very difficult to solve it manually and is often difficult to find a feasible solution when resources are tight. Among the different requirements that are considered in Brazilian Schools, two compactness requirements are essential on a teacher's schedule: to minimize working days and to avoid idle times. In this work we explore the influence of four different idle times constraint formulations: two of them were previously proposed, and two are novel. Experimental results shown that a novel idle times constraint formulation produces better results for classical instances of the literature. One of these results is a new optimal solution and two of them are new best computed solutions. Finally, we also study the impact of double lessons and the minimization of working days in these formulations.

KEYWORDS. School timetabling, Compactness Requirements, Mixed Integer Programming

Combinatorial Optimization, Mathematical Programming

1 Introduction

A common task to all education institutions is the generation of a scheduling of classes that combines teachers, students, rooms and periods (or timeslots) providing a feasible solution. Besides feasible, the solution must be improved as much as possible, attending requirements of different nature. This problem is often solved manually in small and medium sized institutions. However, the automation of this task is becoming more and more common, and it is almost mandatory in large institutions. The reason for that is not only the easiness of having a feasible solution automaticly generated, but the quality of that solution reflects in less time spent generating it, better use of resources and economic gains. This combination results in a better quality of study for the students and work for the teachers.

Requirements are separated into hard and soft ones. By hard requirements we consider those ones that must be satisfied, whilst soft requirements are the ones that may be violeted, but should be satisfied as much as possible. Soft requirements can have different levels of importance that are expressed by a weighted objective function. In general, as many soft requirements are respected, better is the evaluation of the solution.

The timetabling problem first appeared in scientific literature in the 60's (Gotlieb, 1963) and since them it has gaining increasing attention. The most basic problem is to schedule a set of class-teacher events (or meetings) in such a way that no teacher (nor class) is required in more than one lesson at a time. This basic problem can be solved in polynomial time by a min-cost network flow algorithm (de Werra, 1971). However, in a minimal real-world application, teachers can be unavailable in some periods. If this constraint is taken into account, the resulting timetabling problem is NP-complete (Even et al., 1975).

In fact, this problem has several variants proposed in the literature that are NP-complete, and the set of objectives and requirements depends mostly on the context of the application, the school and the place where it is located (Post et al., 2011; Drexl & Salewski, 1997; Schaerf, 1999a). In Brazil, for example, it is very common that a teacher works in more than one school. In order to allow this possibility, it is important to compact the lessons in each school in the minimal number of days. Furthermore, it is required to avoid free periods (idle times) between lessons in a teachers schedule. In addition, due to pedagogical demands or personal preferences, a teacher can request an amount of lessons in the same class conducted in two consecutive periods (double lessons). These set of requirements defines a problem called Class-Teacher Timetabling Problem with Compactness Requirements (CTTPCR). This variant was studied in Souza (2000); Souza et al. (2003); Santos & Ochi (2005); Santos (2007); Bello et al. (2008); Santos et al. (2012).

Among the approaches used to successfully solve this kind of problem, we can highlight metaheuristics as Simulated Annealing (Colorni & Dorigo, 1998; Avella et al., 2007; Zhang et al., 2010), Tabu Search (Schaerf, 1999b; Santos & Ochi, 2005) and Genetic Algorithms (Caldeira & Rosa, 1997). Other advanced techniques also used are Hyper-Heuristcs (Burke et al., 2003) and Constraint Programming (Valouxis & Housos, 2003; Marte, 2007).

For a long time many variations of timetabling have been considered as intractable by exact methods. However, in the last years several improvements have been made in Mixed Integer Programming (MIP) solvers (Lodi, 2010), and therefore motivating studies through this approach (Birbas et al., 2008).

This work extends previous studies in CTTPCR proposing new formulations for the idle times requirement and providing extensive experiments to support conclusions about their impact in the problem solving performance.

This paper is organized as follows. Section 2 formally presents the problem, a base formulation of it and four alternative idle times constraint formulations that can be added to the base formulation. Section 3 presents experimental results considering synthetic and real-world instances. Finally, Section 4 presents conclusions and further work.

2 Problem Description and Modeling

The goal of the CTTPCR, the school timetabling problem that we are considering, is to build a weekly timetable. The week is organized into the set of days D, where each day is splitted in a given set of periods P. We call a timeslot a tuple (d, p), with $d \in D$ and $p \in P$, such that a lesson can be given, assuming that each lesson has the same duration.

Let C be a set of classes and T be a set of teachers. A class $c \in C$ is a group of students that follow the same course and have full availability. Teachers $t \in T$ may be unavailable in some timeslots.

The problem also considers a set of events (or meetings) E such that to each event $e \in E$ is preassigned a class-teacher pair and a given number of lessons (workload) that must be scheduled. In addition, each event defines how lessons are distributed in a week by requesting an amount of double lessons and restricting the daily limit of lessons.

A feasible timetable has an assigned timeslot to each lesson satisfying the hard requirements **H1**-**H5** stated below:

H1 The workload defined in each event must be attended.
H2 A teacher cannot be scheduled to more than one lesson in a given period.
H3 Lessons cannot be taught to the same class in the same period.
H4 A teacher cannot be scheduled to a period in which he/she is unavailable.
H5 The maximum number of daily lessons of each event must be respected.

Besides feasibility, the number of violations of the soft requirements presented below is minimized:

S1 Each teacher has a minimum number of idle times.
S2 Each teacher has a minimum number of working days.
S3 Each event has a minimum number of double lessons.

For each violation on the soft requeriments there is a cost that is embedded in the objective function, weighted accordingly to their importance.

Given this problem, we present the set of additional parameters used by the model:

- S: set of tuples (d, p) for $d \in D$, $p \in P$.
- U: set of tuples (m, n) for $m, n \in P : n > m + 1$.
- Q: set of tuples (m, n) for $m, n \in P : n \geq m < |P| - 1$.
- E_t: set of events a given teacher t is assigned.
- E_c: set of events a given class c is assigned.
- ω_t: cost for each idle time of the teacher t.
- γ_t: cost for each working day of the teacher t. By working day we consider a day that the teacher has at least one lesson assigned to him/her.
- δ_e: cost of each double lesson of event e not taught sequentially.
- R_e: workload of event e
- L_e: maximum daily number of lessons of event e
- $V_{edp} = \begin{cases} 1: & \text{if the teacher assigned to event } e \text{ is available in the timeslot } (d, p), \\ 0: & \text{otherwise.} \end{cases}$
- SG_e: set of periods on which event e can start a double lesson ($SG_e = \{(d, p) \in S : p < |P|, V_{edp} + V_{ed,p+1} = 2\}$)
- MG_e: minimum amount of double lessons required by event e

Next we present the base formulation of the problem.

2.1 Base Formulation

In this section we present a base formulation for the CTTCPR adapted from Santos et al. (2012). We modified this formulation to simplify its presentation replacing each pre-assigned encounters between teacher/class by a set of events. Experimental evaluation demonstrated that to minimize the number of idle times of teachers turns the resolution of the problem considerably more difficult. Thus, we first explore the problem without the soft requirement **S1**. Later we explore the effect of its inclusion in the formulation.

The objective function of the problem formulation consists of three weighted parts related to the soft requirements **S1, S2** and **S3** respectively.

Decision Variables:

- $x_{edp} \in \{0,1\} = \begin{cases} 1: & \text{if the event } e \text{ is scheduled to timeslot } (d,p), \\ 0: & \text{otherwise.} \end{cases}$

Auxiliary Variables:

- $y_{td} \in \{0,1\} = \begin{cases} 1: & \text{if at least one lesson is assigned to teacher } t \text{ on day } d, \\ 0: & \text{otherwise.} \end{cases}$
- $g_{edp} \in \{0,1\} = \begin{cases} 1: & \text{if event } e \text{ has a double lesson starting at timeslot } (d,p), \\ 0: & \text{otherwise.} \end{cases}$
- $G_e \in \mathbb{Z}^+$: count the number of double lessons remaining to reach MG_e.
- $W_t \in \mathbb{Z}^+$: total of idle times of teacher t.

Objective function:

$$\textbf{minimize} \sum_{t \in T} \omega_t W_t + \sum_{t \in T} \sum_{d \in D} \gamma_t y_{td} + \sum_{e \in E} \delta_e G_e \tag{1}$$

subject to:

$$\sum_{(d,p) \in S} x_{edp} = R_e \qquad \forall e \in E \tag{2}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e \in E, d \in D \tag{3}$$

$$x_{edp} \leq V_{edp} \qquad \forall e \in E, (d,p) \in S \tag{4}$$

$$\sum_{e \in E_t} x_{edp} \leq 1 \qquad \forall t \in T, (d,p) \in S \tag{5}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c \in C, (d,p) \in S \tag{6}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t \in T, (d,p) \in S \tag{7}$$

$$g_{edp} \leq x_{edp} \qquad \forall e \in E, (d,p) \in SG_e \tag{8}$$

$$g_{edp} \leq x_{e,d,p+1} \qquad \forall e \in E, (d,p) \in SG_e \tag{9}$$

$$G_e \geq MG_e - \sum_{(d,p) \in SG_e} g_{edp} \qquad \forall e \in E \tag{10}$$

$$\sum_{d \in D} y_{td} \geq \max \left\{ \left\lceil \frac{\sum_{e \in E_t} R_e}{|P|} \right\rceil, \max_{e \in E_t} \left\{ \left\lceil \frac{R_e}{L_e} \right\rceil \right\} \right\} \qquad \forall t \in T \tag{11}$$

$$x_{edp} \in \{0,1\} \qquad \forall e \in E, (d,p) \in S \tag{12}$$

$$g_{edp} \geq 0 \qquad \forall e \in E, (d,p) \in S \tag{13}$$

$$G_e \geq 0 \qquad \forall e \in E, (d,p) \in S \tag{14}$$

$$y_{td} \geq 0 \qquad \forall t \in T, d \in D \tag{15}$$

Constraint set (2) ensures that the workload of each event is fully scheduled. Constraint set (3) provides a daily limit of lessons for each event. Constraint set (4) ensures that the lessons of an event are scheduled in available periods. Constraints sets (5) and (6) ensure that both teacher's events and class's events are scheduled to only one period at a time. Constraint set (7) identifies the working days of teachers. Constraints sets (8) and (9) allow to create double lessons when the variable g_{edp} is activated. By constraint set (10) G_e stores the number of double lessons remaining to reach MG_e. Since G_e is considered into the objective function, the sum in the left side of the inequality tends to increase, and thus double lessons are created.

Constraint set (11) is a cut proposed by Souza (2000) that defines a minimum number of working days to each teacher. This lower bound makes the formulation stronger.

In the next subsections 2.2, 2.5, 2.4 and 2.3 we present four different formulations that take into account the idle times in the scheduling of teachers. Adding the respective constraints to the base formulation, the first sum of the objective function can now assume non-zero values.

2.2 Idle times formulation β_1

This formulation was first presented by Souza (2000) and improved by Santos et al. (2012). We adapted the improved formulation to our problem by using events. The following variables were used:

- $z_{td} \in \mathbb{Z}^+$: number of idle times of teacher t on day d
- $A'_{td} \in \mathbb{Z}^+$: timeslot of the first lesson of teacher t on day d
- $A''_{td} \in \mathbb{Z}^+$: timeslot of the last lesson of teacher t on day d

$$A'_{td} \leq (|P|+1) - (|P|+1-p) * \sum_{e \in E_t} x_{edp} \qquad \forall t \in T, (d,p) \in S \qquad (16)$$

$$A''_{td} \geq p * \sum_{e \in E_t} x_{edp} \qquad \forall t \in T, (d,p) \in S \qquad (17)$$

$$z_{td} \geq A''_{td} - A'_{td} + y_{td} - \sum_{e \in E_t} \sum_{p \in P} x_{edp} \qquad \forall t \in T, d \in D \qquad (18)$$

$$A'_{td}, A''_{td}, z_{td} \geq 0 \qquad \forall t \in T, d \in D \qquad (19)$$

Constraint sets (16) and (17) force A' and A'' store, respectively, the first and the last teaching periods of a teacher. In constraints (18), the expression $A''_{td} - A'_{td} + y_{td}$ defines the interval between the first and the last teaching periods of a teacher in a day. When the teaching periods are subtracted from this interval, z_{td}, the number of idle times is obtained.

Equation (20) defines the value of W_t used in the objective function.

$$W_t = \sum_{d \in D} z_{td} \qquad (20)$$

2.3 Idle times formulation β_2

In this formulation, used in Avella et al. (2007), each idle time is identified individually by inclusion of the variable z_{tdp} into the base formulation.

- $z_{tdp} \in \{0,1\} = \begin{cases} 1 \text{ if timeslot } (d,p) \text{ is an idle time of teacher } t \\ 0 \text{ otherwise} \end{cases}$

$$z_{tdp} \geq -1 + \sum_{e \in E_t} (x_{edm} + x_{edn} - x_{edp}) \qquad \forall t \in T, (d,p) \in S, (m,n) \in U : m < p < n \tag{21}$$

$$z_{tdp} \leq y_{td} \qquad \forall t \in T, (d,p) \in S \tag{22}$$

$$z_{tdp} \geq 0 \qquad \forall t \in T, (d,p) \in S \tag{23}$$

In constraint set (21), when the variables x_{edm} and x_{edn} are activated and the period x_{edp} is idle, thus the variable z_{tdp} is activated. Constraint set (22) disables all variables z_{tdp} on day d according to variable y_{td}. Although this constraint set is not required to ensure the correctness of the formulation, in our experiments it has improved the performance of the solver.

Equation (24) allows us to calculate W_t in the objective function.

$$W_t = \sum_{(d,p) \in S} z_{tdp} \tag{24}$$

2.4 Idle times formulation β_3

This is a novel formulation that we are proposing by small changes in the formulation β_2. The variable z_{tdmn} is added to the base formulation to store a count of idle times.

- $z_{tdmn} \in \mathbb{Z}^+$: idle times of teacher t between timeslots m and n on day d.

$$z_{tdmn} \geq (n-m-1) * \left(-1 + \sum_{e \in E_t} \left(x_{edm} + x_{edn} - \sum_{m<p<n} x_{edp} \right) \right) \qquad \forall t \in T, d \in D, (m,n) \in U \tag{25}$$

$$z_{tdmn} \geq 0 \qquad \forall t \in T, d \in D, (m,n) \in U \tag{26}$$

In constraint set (25), when the variables x_{edm} and x_{edn} are activated and there is no teaching periods between them, thus variable z_{tdmn} assumes the value $(n-m-1)$ which is the number of idle lessons between periods m and n.

Equation (27) defines the value of W_t used in the objective function.

$$W_t = \sum_{d \in D} \sum_{(m,n) \in U} z_{tdmn} \tag{27}$$

2.5 Idle times formulation β_4

This is a novel formulation based in a network-flow that we are proposing in this work. The vertices are the periods on a day and the activated arcs $(m,n) \in Q$ are the number $(n-m-1)$ of idle times. The set Q contains a list of arcs that can be activated.

- $z_{tdmn} \in \{0,1\} = \begin{cases} 1: & \text{if exist an arc } (m,n) \text{ on day } d \text{ for the teacher } t, \\ 0: & \text{otherwise.} \end{cases}$

$$\sum_{(m,n)\in Q} z_{tdmn} = y_{td} \qquad \forall t \in T, (d,m) \in S : m \leq |P|-2 \tag{28}$$

$$z_{tdmn} \leq 1 + \sum_{e\in E_t} (x_{edm+1} - x_{edm}) \qquad \forall t \in T, d \in D, (m,n) \in Q : m = n \tag{29}$$

$$z_{tdmn} \leq 1 - \sum_{e\in E_t} x_{edp} \qquad \forall t \in T, (d,p) \in S, (m,n) \in Q : p > n = m+1 \tag{30}$$

$$z_{tdmn} \leq \sum_{e\in E_t} x_{edn} \qquad \forall t \in T, d \in D, (m,n) \in U \tag{31}$$

$$\sum_{(m,n)\in Q} z_{tdmn} \leq y_{td} \qquad \forall t \in T, (d,n) \in S : n \geq 3 \tag{32}$$

$$z_{tdmn} \in \{0,1\} \qquad \forall t \in T, d \in D, (m,n) \in Q \tag{33}$$

Constraint set (28) ensures that there is exactly one arc leaving each period. Constraint set (29) states that an arc (m,m) can exist when $x_{edm} = 0$ or $x_{edm} + x_{edm+1} = 2$. Constraint set (30) states that an arc $(m,m+1)$ can exist when x_{edm} is the last lesson on day. Constraint set (31) states that an arc (m,n) can exist when $x_{edn} = 1$.

In a sub-optimal solution, this formulation without constraint set (32) can over-estimate the number of idle times as showed in the Figure 1. The sub-optimal solution (a) has cost 4, while the optimal solution (b) costs 2.

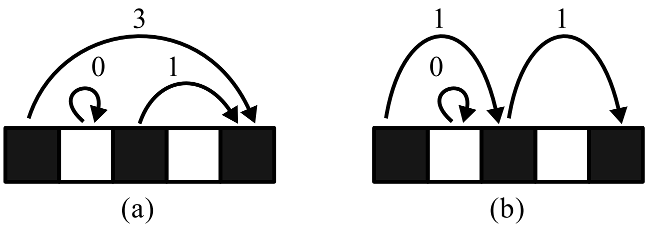

Figure 1: Cost differences between optimal and sub-optimal solutions.

By adding constraint set (32) we exclude sub-optimal solutions of type (a). This constraint set states that there exists at most one arc incoming in each period.

Equation (34) defines the value of W_t used in the objective function.

$$W_t = \sum_{d\in D} \sum_{(m,n)\in U} (n-m-1) * z_{tdmn} \tag{34}$$

3 Computational Experiments

In this section we present experimental evaluation for the mathematical models presented in the previous section, using different schemes built over a different combination of requirements. The scheme μ_0 represents the base model presented at Section 2.1 and it covers all requirements with exception of S1. The scheme μ_1 is equal to μ_0 without S2. The scheme μ_2 is equal to μ_0 without S3. To evaluate the models we used the set of instances available from the repository of *Laboratório de Inteligência Computacional da Universidade Federal Fluminense* (LABIC, 2008). These instances were used by Souza (2000); Souza et al. (2003); Santos & Ochi (2005); Santos (2007); Bello et al. (2008); Santos et al. (2012). Table 1 presents the main characteristics of the instances. In this table, the first column presents the instance identificator, columns 2-4 show the number of days, periods and timeslots, respectively, while that Columns 5-7 present the number of teachers, classes and events, respectively. Finally, the last two columns present the total required double lessons and workload, respectively.

Table 1: Instance characteristics

Inst.	$\lvert D\rvert$	$\lvert P\rvert$	$\lvert S\rvert$	$\lvert T\rvert$	$\lvert C\rvert$	$\lvert E\rvert$	$\sum_{e\in E} MG_e$	$\sum_{e\in E} R_e$
1	5	5	25	8	3	21	21	75
2	5	5	25	14	6	63	29	150
3	5	5	25	16	8	69	4	200
4	5	5	25	23	12	127	66	300
5	5	5	25	31	13	119	71	325
6	5	5	25	30	14	140	63	350
7	5	5	25	33	20	205	84	500

To solve the CTTPCR problem instances, we used the mixed integer programming solver CPLEX 12.1 with default settings. The reported results were computed in a Desktop-PC equipped with an Intel Core i5-2300 processor clocked at 2.8GHz, 4GB of RAM, over a 64 bits Linux operation system. The parameters γ_t, ω_t and δ_e were set to 9, 3 and 1, respectively. These values are the same used by previous works in this problem.

Results presented in tables 2, 3, 4 and 5 were reported with a time limit of 60 minutes. Column *time* shows the running time of the solver in minutes. Column *obj* shows the value of the objective function. Column *LB* shows the lower bound found by the solver. Column *gap* presents the percentual deviation between the best found solution and the lower bound computed as $(obj - LB)/(obj * 100)$. Columns *rows* and *cols* present, respectively, the number of constraints and variables after pre-processing phase of the solver. Column *nodes* shows the number of explored nodes through the whole search. Finally, column *root* shows the time in seconds for solving the linear relaxation at the root node. Best results are showed in bold.

The goal of the first experiment was to solve the CTTPCR without taking into account the idle times (**S1**). Table 2 presents results for model μ_0. As expected, running time increases with the instance size. Nevertheless, all instances were solved optimally in at most 60 minutes.

In the second experiment we inserted the idle times formulations into the model μ_0 obtaining a full model to solve the CTTPCR. In Table 3 are the reported results. Unlike the previous experiment, it was not possible to find optimality for any instance. Formulation β_4 produces better solutions for most of the instances.

Table 2: Results for Formulation μ_0

inst	time (min)	obj	LB	gap(%)	rows	cols	nodes	root(s)
1	0.0	189	189	0.00	831	715	236	0.0
2	0.1	333	333	0.00	1853	2109	0	0.2
3	0.1	414	414	0.00	961	1528	88	0.1
4	1.9	643	643	0.00	1983	2984	531	0.5
5	13.4	756	756	0.00	4139	4417	1297	1.6
6	13.9	738	738	0.00	4173	4928	866	1.9
7	54.6	999	999	0.00	5527	7054	926	4.4
Avg.	12.0	582	582	0.00	2781	3390	563	1.3

The goal of the third experiment is to solve the problem without taking into account the minimization of working days (**S2**). In Table 4, independently of which idle time formulation is used, some instances could be solved optimally. Again, the formulation β_4 produced better solutions than the others.

The fourth experiment evaluates the problem without taking into account the double lessons (**S3**). In Table 5, almost all final solutions are sub-optimal. The results show that among the soft requirements, double lessons requirement leads to less influence on the resolution process. However, when dismissing it, we found better solutions for the idle times formulation β_2, instead of β_4.

Figure 2 presents a graphical comparison of idles time formulations considered in this work.

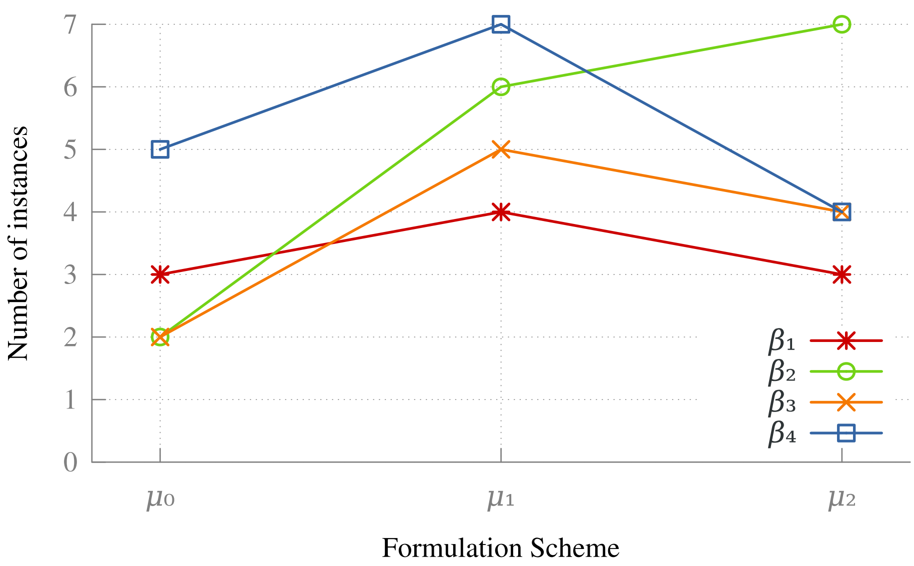

Figure 2: Number of instances with best objective value for $\mu_i \cup \beta_j$

The best known results for the same instance set evaluated in this work are presented in Santos et al. (2012). These results were reached by very long runs of a set of heuristic methods and there is no precise information about the time spent. In Table 6 we present a comparison of the results between the proposed model $\mu_0 \cup \beta_4$ against results reproduced from Santos et al. (2012). Column *LB* presents the best known lower bounds computed by a column generation algorithm in a separated experiment performed by Santos et al. (2012). These values are used to calculate the column gap of both solutions. Columns *CLB*, *cgap*, *last* and *time* present the lower bound and gap found by CPLEX, the time which the last solution was found and the solver runtime. $t.l.$ indicates that the

Table 3: Comparison results for model μ_0

	$\mu_0 \cup \beta_1$							$\mu_0 \cup \beta_2$							$\mu_0 \cup \beta_3$							$\mu_0 \cup \beta_4$						
inst	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root
1	**60.0**	6.44	**202**	1191	**811**	313263	**0.1**	**60.0**	6.44	**202**	1183	**811**	193590	**0.1**	**60.0**	6.44	**202**	**1023**	**811**	**537049**	**0.1**	**60.0**	**6.40**	**202**	1491	1099	273744	**0.1**
2	60.0	2.06	340	2700	**2304**	90890	0.8	60.0	1.77	339	2568	**2304**	73312	**0.5**	60.0	2.06	340	**2243**	**2304**	**130049**	**0.5**	**41.0**	**0.00**	**333**	3210	2889	36614	0.6
3	**60.0**	**2.82**	**426**	1665	**1720**	93516	0.5	**60.0**	**2.82**	**426**	1665	**1720**	102630	**0.3**	**60.0**	3.50	429	**1345**	**1720**	**144810**	**0.3**	**60.0**	**2.82**	**426**	2229	2296	125094	**0.3**
4	**60.0**	1.68	654	3075	3250	24697	1.6	**60.0**	1.83	655	2815	**3215**	19492	1.2	**60.0**	2.72	661	**2439**	**3215**	**30767**	**1.0**	**60.0**	**1.38**	**652**	3491	3906	20599	1.1
5	**60.0**	4.67	793	5979	4882	5227	4.1	**60.0**	2.45	775	5789	**4867**	5858	3.0	**60.0**	**2.33**	**774**	**5039**	**4867**	**12131**	**2.7**	**60.0**	5.62	801	7299	6277	8099	3.4
6	**60.0**	8.55	807	6196	**5372**	1407	5.5	**60.0**	6.58	790	5801	**5372**	3472	3.8	**60.0**	6.58	790	**5061**	**5372**	**8416**	**3.0**	**60.0**	**5.14**	**778**	7300	6704	4170	3.6
7	**60.0**	**13.51**	**1155**	7937	**7549**	514	12.6	**60.0**	21.59	1274	7342	**7549**	525	7.7	**60.0**	15.70	1185	**6517**	**7549**	**593**	**7.4**	**60.0**	20.65	1259	8962	9034	524	8.8
Avg.	60.0	5.68	**625**	4106	3698	75644	3.6	60.0	6.21	637	3880	**3691**	56982	2.4	60.0	**5.62**	626	**3381**	**3691**	**123402**	**2.1**	**57.3**	6.00	636	4854	4600	66977	2.6
Best	6	2	3	0	5	0	1	6	1	2	0	**7**	0	3	6	1	2	**7**	**7**	**7**	**7**	**7**	**5**	**5**	0	0	0	2

Table 4: Comparison results for model without requirement **S2**

	$\mu_1 \cup \beta_1$							$\mu_1 \cup \beta_2$							$\mu_1 \cup \beta_3$							$\mu_1 \cup \beta_4$						
inst	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root
1	1.6	**0.00**	**0**	1111	**771**	**14500**	0.1	**0.1**	**0.00**	**0**	1143	**771**	681	**0.0**	0.8	**0.00**	**0**	**1015**	**771**	8605	**0.0**	**0.1**	**0.00**	**0**	1303	1059	447	0.1
2	5.5	**0.00**	**0**	2424	**2236**	**11211**	0.6	**0.1**	**0.00**	**0**	2489	**2236**	0	0.4	2.0	**0.00**	**0**	**2229**	**2236**	1954	**0.3**	**0.1**	**0.00**	**0**	2814	2821	0	0.4
3	26.5	**0.00**	**0**	1521	**1640**	58247	0.3	2.0	**0.00**	**0**	1585	**1640**	795	0.2	60.0	100.00	3	**1329**	**1640**	**203365**	**0.1**	**0.5**	**0.00**	**0**	1905	2216	528	0.2
4	30.2	**0.00**	**4**	2705	3174	**10716**	1.1	5.8	**0.00**	**4**	2727	**3149**	620	**0.9**	5.2	**0.00**	**4**	**2427**	**3149**	1085	**0.9**	**4.4**	**0.00**	**4**	3133	3838	909	**0.9**
5	60.0	100.00	12	5371	4742	3061	3.4	**1.7**	**0.00**	**0**	5436	**4727**	17	**2.1**	27.6	**0.00**	**0**	**4836**	**4727**	**3352**	**2.1**	10.9	**0.00**	**0**	6366	6137	771	2.4
6	60.0	100.00	21	5475	**5222**	3735	4.0	**0.3**	**0.00**	**0**	5623	**5222**	0	3.1	53.4	**0.00**	**0**	**5031**	**5222**	**8007**	**2.2**	2.1	**0.00**	**0**	6363	6554	24	2.6
7	**60.0**	**100.00**	99	6979	**7384**	540	8.4	**60.0**	**100.00**	34	7144	**7384**	550	6.8	**60.0**	**100.00**	156	**6484**	**7384**	**626**	6.8	**60.0**	**100.00**	**33**	7969	8869	606	**5.4**
Avg.	34.8	42.86	19	3655	3595	14572	2.5	**10.0**	**14.29**	**5**	3735	**3589**	380	1.9	29.8	28.57	23	**3335**	**3589**	**32427**	1.8	11.1	**14.29**	**5**	4264	4499	469	**1.7**
Best	1	5	4	0	5	3	0	**5**	**7**	6	0	**7**	0	3	1	6	5	**7**	**7**	**4**	**6**	**5**	**7**	**7**	0	0	0	2

Table 5: Comparison results for model without requirement **S3**

	$\mu_2 \cup \beta_1$							$\mu_2 \cup \beta_2$							$\mu_2 \cup \beta_3$							$\mu_2 \cup \beta_4$						
inst	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root	time	gap	obj	rows	cols	nodes	root
1	**60.0**	**5.97**	**201**	696	**556**	**3919893**	**0.0**	**60.0**	**5.97**	**201**	688	**556**	2167417	**0.0**	**60.0**	**5.97**	**201**	**528**	**556**	1929407	**0.0**	**60.0**	**5.97**	**201**	996	844	3820818	**0.0**
2	45.1	**0.00**	**333**	1567	**1723**	**127521**	0.3	**1.0**	**0.00**	**333**	1435	**1723**	1004	**0.2**	3.9	**0.00**	**333**	**1110**	**1723**	10733	**0.2**	6.7	**0.00**	**333**	2077	2308	26924	**0.2**
3	**60.0**	3.50	429	1533	**1652**	71634	0.5	**60.0**	**2.13**	**423**	1533	**1652**	89086	**0.2**	**60.0**	**2.13**	**423**	**1213**	**1652**	**204179**	**0.2**	**60.0**	2.82	426	2097	2228	129222	**0.2**
4	**60.0**	**1.39**	**648**	2203	2802	59831	0.8	**60.0**	**1.39**	**648**	1943	**2767**	52611	**0.5**	**60.0**	**1.39**	**648**	**1567**	**2767**	**81551**	**0.5**	**60.0**	**1.39**	**648**	2619	3458	74644	0.6
5	**60.0**	1.95	771	3437	3580	23893	1.8	**60.0**	**0.79**	**762**	3247	**3565**	25224	1.0	**60.0**	2.33	774	**2497**	**3565**	**48368**	**0.8**	**60.0**	**0.79**	**762**	4757	4975	28111	1.2
6	**60.0**	3.91	768	3613	**4049**	11934	2.2	**60.0**	**3.15**	**762**	3218	**4049**	16268	1.2	**60.0**	5.02	777	**2478**	**4049**	19319	**1.0**	**60.0**	3.53	765	4717	5381	**21332**	1.2
7	**60.0**	4.03	1041	4493	**5785**	2431	4.7	**60.0**	**2.06**	**1020**	3898	**5785**	3520	3.0	**60.0**	2.63	1026	**3073**	**5785**	**6178**	2.6	**60.0**	4.03	1041	5518	7270	3887	**2.4**
Avg.	57.9	2.96	599	2506	2878	**602448**	1.5	**51.6**	**2.21**	**593**	2280	**2871**	336447	0.9	52.0	2.78	597	**1780**	**2871**	328533	**0.8**	52.4	2.65	597	3254	3780	586419	**0.8**
Best	6	3	3	0	5	2	1	**7**	**7**	**7**	0	**7**	0	4	6	4	4	**7**	**7**	**4**	**6**	6	4	4	0	0	1	4

time limit has been reached.

Results found by CPLEX with model $\mu_0 \cup \beta_4$, with time limit of 10 hours, are equal or slightly better than the best known results, except for instance 6. The solution found to instance 4 is a new known optimal solution. We also found new best known solutions for instances 5 and 7. Although the solver has found the optimal solutions for instances 1-4 in less than 5000 seconds, it proves the optimality only for the instance 3. In fact, the instance 3 is the only one that has the lower bound provided by CPLEX equal to optimal objective value.

Table 6: Comparison of model $\mu_0 \cup \beta_4$ against best known results

		best known		$\mu_0 \cup \beta_4$					
Inst.	LB	obj	gap	obj	CLB	gap	cgap	last	time
1	202	**202**	0.0	**202**	189	0.0	6.4	2390	t.l.
2	333	**333**	0.0	**333**	333	0.0	0.0	2457	2457
3	423	**423**	0.0	**423**	414	0.0	2.1	4817	t.l.
4	652	653	0.2	**652**	643	0.0	1.4	1753	t.l.
5	762	766	0.5	**764**	756	0.3	1.1	33147	t.l.
6	756	**760**	0.5	765	738	1.2	3.5	33431	t.l.
7	1017	1029	1.2	**1028**	999	1.1	2.8	33992	t.l.

4 Conclusion

In this work we present model formulations for the Class-Teacher Timetabling Problem with Compactness Requirements considering four different idle times formulations. Additionally we performed computational experiments to evaluate the impact of soft requirements in the resolution process. Experimental results show that the idle times requirement is the compactness requirement that most aggregates complexity to the resolution of problem by CPLEX. The requirement of double lessons is the one that influences less in the resolution process. The main contribution of this paper is the novel formulation model $\mu_0 \cup \beta_4$ that outperforms the previously proposed ones when solving CTTPCR. As it could provide good solutions within 60 minutes, it is reasonable to state that this approach of mixed integer programming can be used to build school timetables in practice. As additional contribution, we compared the experimental results with previous studies and we found one new optimal solution and two new best computed solutions for the instance set evaluated.

Acknowledgements

This work was partially supported by CAPES (Coordination for the Improvement of Higher Level Personnel) and Petrobras S.A.,Brazil.

References

Avella, P., Salerno, S., & Vasil, I. (2007). A computational study of local search algorithms for Italian high-school timetabling. *Journal of Heuristics*, pages 543–556.

Bello, G., Rangel, M., & Boeres, M. (2008). An approach for the class/teacher timetabling problem using graph coloring. *The Practice and Theory of Automated Timetabling VII*.

Birbas, T., Daskalaki, S., & Housos, E. (2008). School timetabling for quality student and teacher schedules. *Journal of Scheduling*, 12(2):177–197.

Burke, E., Kendall, G., & Soubeiga, E. (2003). A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics*, 9(6):451–470.

Caldeira, J. & Rosa, A. (1997). School timetabling using genetic search. *Practice and Theory of Automated Timetabling, Toronto*.

Colorni, A. & Dorigo, M. (1998). Metaheuristics for high school timetabling. *Computational optimization and*, 298:275–298.

de Werra, D. (1971). Construction of school timetables by flow methods. *INFOR*, I(9):12–22.

Drexl, A. & Salewski, F. (1997). Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, 102(1):193–214.

Even, S., Itai, A., & Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on*, pages 184–193. IEEE.

Gotlieb, C. (1963). The construction of class-teacher timetables. In *IFIP*, pages 73–77, Amsterdam. North-Holland.

LABIC (2008). Instances of school timetabling, `http://labic.ic.uff.br`.

Lodi, A. (2010). Mixed integer programming computation. *50 Years of Integer Programming 1958-2008*, pages 619–645.

Marte, M. (2007). Towards constraint-based school timetabling. *Annals of Operations Research*, 155(1):207–225.

Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., & Schaerf, A. (2011). XHSTT: an XML archive for high school timetabling problems in different countries. *Annals of Operations Research*.

Santos, H. & Ochi, L. (2005). A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *Journal of Experimental Algorithmics*, 10(2):1–16.

Santos, H. G. (2007). *Formulações e Algoritmos para o Problema de Programação de Horários em Escolas*. h7, Universidade Federal Fluminense.

Santos, H. G., Uchoa, E., Ochi, L. S., & Maculan, N. (2012). Strong bounds with cut and column generation for class-teacher timetabling. *Annals of Operations Research*.

Schaerf, A. (1999a). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127.

Schaerf, A. (1999b). Local search techniques for large high school timetabling problems. *systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 29(4):368–377.

Souza, M. (2000). *Programação de horários em escolas: uma aproximação por metaheuristicas*. Doctoral Thesis, Universidade Federal do Rio de Janeiro.

Souza, M., Ochi, L., & Maculan, N. (2003). A GRASP-Tabu search algorithm for solving school timetabling problems. *Metaheuristics: Computer Decision-Making. Kluwer Academic Publishers, Boston*, pages 659–672.

Valouxis, C. & Housos, E. (2003). Constraint programming approach for school timetabling. *Computers & Operations Research*, 30(10):1555–1572.

Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. C. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203(3):550–558.