World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:7, 2015

Joint Training Offer Selection and Course Timetabling Problems: Models and Algorithms

Gianpaolo Ghiani, Emanuela Guerriero, Emanuele Manni, Alessandro Romano

International Science Index, Computer and Information Engineering Vol:9, No:7, 2015 waset.org/Publication/10002234

Abstract—In this article, we deal with a variant of the classical course timetabling problem that has a practical application in many areas of education. In particular, in this paper we are interested in high schools remedial courses. The purpose of such courses is to provide under-prepared students with the skills necessary to succeed in their studies. In particular, a student might be under prepared in an entire course, or only in a part of it. The limited availability of funds, as well as the limited amount of time and teachers at disposal, often requires schools to choose which courses and/or which teaching units to activate. Thus, schools need to model the training offer and the related timetabling, with the goal of ensuring the highest possible teaching quality, by meeting the above-mentioned financial, time and resources constraints. Moreover, there are some prerequisites between the teaching units that must be satisfied. We first present a Mixed-Integer Programming (MIP) model to solve this problem to optimality. However, the presence of many peculiar constraints contributes inevitably in increasing the complexity of the mathematical model. Thus, solving it through a general-purpose solver may be performed for small instances only, while solving real-life-sized instances of such model requires specific techniques or heuristic approaches. For this purpose, we also propose a heuristic approach, in which we make use of a fast constructive procedure to obtain a feasible solution. To assess our exact and heuristic approaches we perform extensive computational results on both real-life instances (obtained from a high school in Lecce, Italy) and randomly generated instances. Our tests show that the MIP model is never solved to optimality, with an average optimality gap of 57%. On the other hand, the heuristic algorithm is much faster (in about the 50% of the considered instances it converges in approximately half of the time limit) and in many cases allows achieving an improvement on the objective function value obtained by the MIP model. Such an improvement ranges between 18% and 66%.

Keywords—Heuristic, MIP model, Remedial course, School, Timetabling.

I. Introduction

In this paper, we study a variant of the classical course timetabling problem. More specifically, we focus on remedial courses that have a practical application in many areas of education. Remedial courses aim at providing under-prepared students the skills they need to succeed in their studies. A great restriction when organizing such courses is the limited availability of funds and teachers, thus forcing institutions to choose which courses to activate. The goal is to model a suitable training offer and a related timetabling satisfying the largest possible number of students with the highest possible teaching quality. At the same time, they must meet the above-mentioned financial and resources constraints. The originality and the complexity of the approach we propose for this problem can be found in the following elements of novelty:

- economic constraints;
- timing constraints;
- prerequisite constraints between teaching units.

The remainder of the paper is organized as follows. In Section II we review the relevant literature, whereas in Sections III and IV we present a mathematical formulation and a heuristic approach, respectively. Then, in Section V we describe our computational experiments on both real and randomly-generated instances. Finally, conclusions follow in Section VI.

G. Ghiani is with the Department of Engineering for Innovation, University of Salento, Lecce, LE, 73100 Italy, e-mail: gianpaolo.ghiani@unisalento.it.
E. Guerriero is with the Department of Engineering for Innovation, University of Salento, Lecce, LE, 73100 Italy, e-mail: emanuela.guerriero@unisalento.it.
E. Manni is with the Department of Engineering for Innovation, University of Salento, Lecce, LE, 73100 Italy, e-mail: emanuele.manni@unisalento.it.
A. Romano is with the Department of Engineering for Innovation, University of Salento, Lecce, LE, 73100 Italy, e-mail: alessandro.romano@unisalento.it.

II. Literature review

The problem we study has some similarities with the problem of course timetabling, which is broadly studied in the literature. However, our problem differs substantially for the presence of constraints that preserve the teaching quality, like prerequisite constraints that determine the possible sequence of teaching units, for the funds and timing constraints that impose a selection of the contents to delivery and for a different objective function that aims at satisfying the largest number of students, because there is no guarantee that all teaching units will be provided. The model educational timetabling problem has always aroused the curiosity of the scientific community related to the world of operations research. In the following, we overview the main methods used to solve this problem. Among the oldest articles there are those using Integer Programming (IP) ([1],[2]), even if in the real cases, the huge amount of data requires alternative solutions to reduce the complexity of the IP model. In particular, [3] propose a possible decomposition in several stages of the IP model in order to reduce its complexity. In [4], the authors underline that a mixed-integer programming model is not appropriate for practical-sized test instances and propose two decomposition approaches. The first is based on a two-stage modeling solution approach, where in the initial stage weekly time-slots for the classes are determined, while in the second stage teachers are assigned to classes. The second approach is based on another mixed-integer programming formulation that selects valid combinations of weekly schedules from the set of all feasible schedules, and uses a column generation framework to exploit its inherent special structure.

Recently, many meta-heuristic techniques have been proposed. Among the best known meta-heuristics paradigms there is the Tabu Search (TS) used in a classic manner in [5] and in a more innovative manner in [6], because it integrates it with several distinguished features such as an original double Kempe chains neighborhood structure, a penalty guided perturbation operator and an adaptive search mechanism. In [7] TS is hybridized with a genetic algorithm and it is applied on a set of neighborhood structures. Among the others meta-heuristics, well known is also the Simulated Annealing (SA) used in [8] to improve an initial solution obtained through an approach of mathematical programming combined with Lagrangian relaxation. In [9] SA is used in a more innovative manner because the search process is applied on variable neighbourhood. In addition to the previous meta-heuristics there are Genetic Algorithms (GA) that are used in [10] to optimize the initial solution through a fitness function. In [11] GA are combined with a search strategy to create offspring into population based on a data structure that stores information extracted from good individuals of previous generations and with a local search that improves the efficiency of the proposed GA. In [12] the approach is based on Ant Colony Optimization, where artificial ants successively construct solutions based on *pheromones* and local information. The key feature is the use of two distinct but simplified pheromone matrices in order to improve convergence, still providing enough flexibility for effectively guiding the solution construction process. By parallelizing the algorithm it is possible to improve the solution quality significantly. Alternative solutions contemplate the use of Constraint Programming as in [13].

III. Problem formulation

Before presenting our mathematical model, we first introduce the following notation:

- S is the set of students having one or more teaching units to recovery;
- C is the set of all courses to recovery;
- U_{rec} is the set of teaching units to recovery;
- v_{su} represents the result obtained by student $s \in S$ concerning teaching unit $u \in U_{rec}$. If student s must recovery teaching unit u, then such a coefficient is equal to 1. Otherwise it is equal to 0;
- f_{cu} indicates if teaching unit $u \in U_{rec}$ is present in course $c \in C$. In this case, the coefficient is equal to 1. Otherwise, it is equal to 0;
- p_{ij} indicates whether teaching unit $j \in U_{rec}$ is a prerequisite for teaching unit $i \in U_{rec}$. The possible values are 1 or 0;
- TS is the set of all available timeslots;
- D is the set of all available teachers;
- a_{dc} indicates whether teacher $d \in D$ has the appropriate skills to teach course $c \in C$;
- r_u is the number of students that must recovery teaching unit $u \in U_{rec}$;
- l_u is the number of timeslots required by teaching unit $u \in U_{rec}$;
- G is the set of all available days;
- W is the set of all available weeks;
- h is the number of daily timeslots;
- w is the number of weekly timeslots;
- td is the maximum number of daily timeslots per course;
- tw is the maximum number of weekly timeslots per course;
- B is the maximum number of timeslots allowed by the available funds.

The decision variables are:

- $y_{ts,ud}$ binary variable equal to 1 if one lesson of teaching unit ud is held in timeslot ts, otherwise it is equal to 0.
- $z_{d,c}$ binary variable equal to 1 if teacher d is assigned to course c, otherwise it is equal to 0.
- $t_{ts,d,c}$ binary variable equal to 1 if timeslot ts is assigned to teacher d and course c, otherwise it is equal to 0.

The constraints of the model are:

- Each course can be assigned at most to a single teacher qualified to teach it:

$$\sum_{d \in D | a_{dc}=1} z_{d,c} \leq 1 \quad \forall c \in C \tag{1}$$

- If a teaching unit u is held in a timeslots ts, then the course c containing the teaching unit u must be assigned to a teacher qualified to teach it:

$$\begin{gathered} y_{ts,u} \leq \sum_{d \in D | a_{dc}=1} z_{d,c} \\ \forall ts \in TS, \forall c \in C, \forall u \in U_{rec} | f_{cu} = 1 \end{gathered} \tag{2}$$

- No more than one teaching unit of the same course is assigned to each timeslot ts:

$$\sum_{u \in U_{rec} | f_{cu}=1} y_{ts,u} \leq 1 \quad \forall ts \in TS, \forall c \in C \tag{3}$$

- Variables $t_{ts,d,c}$ are tied to variables $z_{d,c}$:

$$t_{ts,d,c} \leq z_{d,c} \quad \forall ts \in TS, \forall d \in D, \forall c \in C \tag{4}$$

- For each timeslot ts, if no teaching unit u of course c is assigned to it, then the corresponding variable $t_{ts,d,c}$ must be equal to 0:

$$\begin{gathered} t_{ts,d,c} \leq \sum_{u \in U_{rec} | f_{cu}=1} y_{ts,u} \\ \forall ts \in TS, \forall d \in D, \forall c \in C \end{gathered} \tag{5}$$

- This constraint models the relationship between decision variables $t_{ts,d,c}$, decision variables $y_{ts,u}$ and decision variables $z_{d,c}$:

$$\begin{gathered} t_{ts,d,c} \geq z_{d,c} + (\sum_{u \in U_{rec} | f_{cu}=1} y_{ts,u}) - 1 \\ \forall ts \in TS, \forall d \in D, \forall c \in C \end{gathered} \tag{6}$$

- Teacher cannot teach more than one lesson in the same timeslot:

$$\sum_{c \in C | a_{dc}=1} t_{ts,d,c} \leq 1 \quad \forall d \in D, \forall ts \in TS \tag{7}$$

– For each teaching unit $u \in U_{rec}$, there is a maximum number of timeslots:

$$\sum_{ts \in TS} y_{ts,u} \leq l_u \quad \forall u \in U_{rec} \tag{8}$$

– A teacher can teach only courses for which he/she is qualified:

$$z_{d,c} * (1 - a_{dc}) = 0 \quad \forall c \in C, \forall d \in D \tag{9}$$

– Students cannot attend more than one lesson in the same timeslot:

$$\sum_{u \in U_{rec} | v_{su}=1} y_{ts,u} \leq 1 \quad \forall s \in S, \forall ts \in TS \tag{10}$$

– The budget must be respected:

$$\sum_{u \in U_{rec}} \sum_{ts \in TS} y_{ts,u} \leq B. \tag{11}$$

– Every teaching unit can be assigned only after its prerequisite teaching units:

$$\begin{aligned} & y_{ts_i,i} + y_{ts_j,j} \leq 1 \\ & \forall i,j \in U_{rec} | p_{ij} = 1, \forall ts_i, ts_j \in TS | ts_i \leq ts_j \end{aligned} \tag{12}$$

– If a teaching unit i is assigned to a timeslot ts_i then at least one timeslot must be assigned to all its prerequisite teaching units:

$$\begin{aligned} & \sum_{ts_j \in TS | ts_j < ts_i} y_{ts_j,j} \geq y_{ts_i,i} \\ & \forall i,j \in U_{rec} | p_{ij} = 1, \forall ts_i \in TS \end{aligned} \tag{13}$$

– Each course can be assigned to a maximum number of timeslots per day:

$$\begin{aligned} & \sum_{u \in U_{rec} | f_{cu}=1} \sum_{\substack{ts=g*h+1 \\ ts \in TS}}^{g*h+h} y_{ts,u} \leq td \\ & \forall g \in G, \forall c \in C \end{aligned} \tag{14}$$

– Each course can be assigned to a maximum number of timeslots per week:

$$\begin{aligned} & \sum_{u \in U_{rec} | f_{cu}=1} \sum_{\substack{ts=w*k+1 \\ ts \in TS}}^{w*k+k} y_{ts,u} \leq tw \\ & \forall w \in W, \forall c \in C \end{aligned} \tag{15}$$

– If in the same day there are two lessons of the same course, then they must to be consecutive:

$$\begin{aligned} & y_{ts_1,u_1} + y_{ts_2,u_2} \leq 1 \\ & \forall g \in G, \forall ts_1, ts_2 \in TS \mid \\ & (ts_2 - ts_1 > 1 \lor ts_2 - ts_1 < -1) \\ & \land (ts_1 > g*h \land ts_1 \leq g*h+h) \\ & \land (ts_2 > g*h \land ts_2 \leq g*h+h), \\ & \forall u_1, u_2 \in U_{rec} \mid \exists\, c \in C \land f_{cu_1} = 1 \land f_{cu_2} = 1 \end{aligned} \tag{16}$$

Our MIP model aims at satisfying as much as possible the recovery needs of the students. To achieve this objective, it is necessary to reduce as much as possible the gap between the number of timeslots needed by a single teaching unit and the timeslots actually assigned to it. In addition, because we want to satisfy as many students as possible, we assign a weight to a single teaching unit gap related to the number of students that need to recovery that specific teaching unit.

$$\min f = \sum_{u \in U_{rec}} r_u \left(l_u - \sum_{ts \in TS} y_{ts,u} \right) \tag{17}$$

IV. Heuristic approach

Solving the problem with the MIP model through a general purpose solver may be performed for small instances only, while solving real-life-sized instances of such model requires specific heuristic approaches. In particular, we have performed a number of experiments based on both real and randomly-generated instances. We have used the general purpose MIP solver IBM ILOG CPLEX and we have imposed a time limit of $3,600$ seconds. In our tests the problem was never solved to optimality and the average optimality gap was equal to about 57%. Moreover, we underline that in the realistic case under examination of a high school, it is very unlikely that the school administration has an hardware so powerful such as that used in our tests (described in Section V).

For this purpose, we propose a heuristic approach based on the use of a fast constructive procedure. Our approach is based on solving a number of sub-MIP problems. The algorithm begins by solving the first sub-MIP problem associated to the first day of the planning horizon, obtaining a partial solution that will become fixed in the following sub-MIP problems. Then, the algorithm solves the sub-MIP problem associated to the first two days (the solution associated to the first day is fixed) determining a new partial solution that will be used in the following sub-MIP problems. Thus, at each step the time window is increased by one day (Fig. 1). The algorithm execution terminates in the following cases:

– we have considered the entire planning horizon;
– we have completely utilized the available funds;
– the objective function was not improved for a given number of iterations;
– a time limit is reached.

During each step, only some of the $y_{ts,u}$ variables are fixed, whereas the $z_{d,c}$ and $t_{ts,d,c}$ variables are always considered as free.

Below, we describe the sub-MIP model we use in our heuristic, by first introducing the following additional notation:

– $y'_{ts,ud}$ is a coefficient taking value 1 if teaching unit ud is assigned to timeslot ts, 0 otherwise;
– TSR is the set of timeslots for which the decision variables $y_{ts,ud}$ are "fixed";
– TSU is the set of timeslots for which the decision variables $y_{ts,ud}$ are "free".

The sub-problem constraints are derived from the initial MIP model, apart for the following constraints:

– If a teaching unit u is held in a timeslots ts, then the course c containing the teaching unit u must be assigned

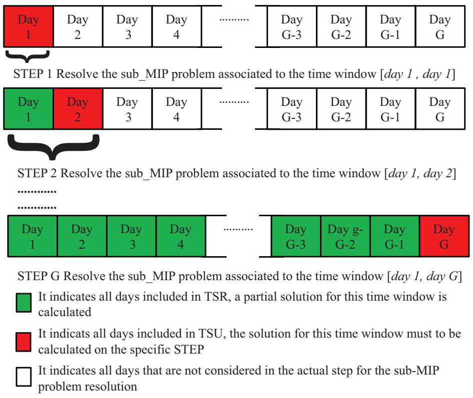

Fig. 1. Time window evolution during the various iterations of the algorithm

to a teacher qualified to teach that course c and constraint (2) becomes:

$$y'_{ts,u} \leq \sum_{d \in D | a_{dc}=1} z_{d,c} \quad \forall ts \in TSR, \forall c \in C, \forall u \in U_{rec}|f_{cu} = 1 \tag{18}$$

$$y_{ts,u} \leq \sum_{d \in D | a_{dc}=1} z_{d,c} \quad \forall ts \in TSU, \forall c \in C, \forall u \in U_{rec}|f_{cu} = 1 \tag{19}$$

– In the same timeslot we do not assign more than one teaching unit related to the same course and constraint (3) becomes:

$$\sum_{u \in U_{rec}|f_{cu}=1} y_{ts,u} \leq 1 \quad \forall ts \in TSU, \forall c \in C \tag{20}$$

– Variables $t_{ts,d,c}$ are tied to variables $z_{d,c}$ and constraint (4) becomes: :

$$t_{ts,d,c} \leq z_{d,c} \quad \forall ts \in \{TSR \cup TSU\}, \forall d \in D, \forall c \in C \tag{21}$$

– For each timeslot ts, if no teaching unit u of course c is assigned to a timeslot ts then the corresponding variable $t_{ts,d,c}$ must be equal to 0 and constraint (5) becomes:

$$t_{ts,d,c} \leq \sum_{u \in U_{rec}|f_{cu}=1} y'_{ts,u} \quad \forall ts \in TSR, \forall d \in D, \forall c \in C \tag{22}$$

$$t_{ts,d,c} \leq \sum_{u \in U_{rec}|f_{cu}=1} y_{ts,u} \quad \forall ts \in TSU, \forall d \in D, \forall c \in C \tag{23}$$

– This constraint models the relationship between decision variables $t_{ts,d,c}$, $y_{ts,u}$ and $z_{d,c}$; constraint (6) becomes:

$$t_{ts,d,c} \geq z_{d,c} + \left(\sum_{u \in U_{rec}|f_{cu}=1} y'_{ts,u} \right) - 1 \quad \forall ts \in TSR, \forall d \in D, \forall c \in C \tag{24}$$

$$t_{ts,d,c} \geq z_{d,c} + \left(\sum_{u \in U_{rec}|f_{cu}=1} y_{ts,u} \right) - 1 \quad \forall ts \in TSU, \forall d \in D, \forall c \in C \tag{25}$$

– Teacher cannot teach more than one lesson in the same timeslot and constraint (7) becomes:

$$\sum_{c \in C | a_{dc}=1} t_{ts,d,c} \leq 1 \quad \forall d \in D, \forall ts \in \{TSR \cup TSU\} \tag{26}$$

– For each teaching unit we cannot assign more than the maximum number of timeslots needed by it and constraint (8) becomes:

$$\sum_{ts \in TSU} y_{ts,u} \leq l_u - \sum_{ts \in TSR} y'_{ts,u} \quad \forall u \in U_{rec} \tag{27}$$

– Students cannot attend simultaneously more than one lesson and constraint (10) becomes:

$$\sum_{u \in U_{rec}|v_{su}=1} y_{ts,u} \leq 1 \quad \forall s \in S, \forall ts \in TSU \tag{28}$$

– The budget cannot be exceeded and constraint (11) becomes:

$$\sum_{u \in U_{rec}} \sum_{ts \in TSU} y_{ts,u} \leq B - \sum_{u \in U_{rec}} \sum_{ts \in TSR} y'_{ts,u} \tag{29}$$

– Every teaching unit must be chronologically assigned only after its prerequisite teaching units and the constraint (12) becomes:

$$y'_{ts_i,i} + y_{ts_j,j} \leq 1 \quad \forall i,j \in U_{rec}|p_{ij} = 1, \quad \forall ts_i \in TSR, \forall ts_j \in TSU | ts_i \leq ts_j \tag{30}$$

$$y_{ts_i,i} + y_{ts_j,j} \leq 1 \quad \forall i,j \in U_{rec}|p_{ij} = 1, \quad \forall ts_i, ts_j \in TSU | ts_i \leq ts_j \tag{31}$$

– If a teaching unit i is assigned to a timeslot ts_i, then at least one timeslot must be assigned to all its prerequisite teaching units and constraint (13) becomes:

$$\sum_{ts_j \in TSR | ts_j < ts_i} y'_{ts_j,j} + \sum_{ts_j \in TSU | ts_j < ts_i} y_{ts_j,j} \geq y_{ts_i,i} \quad \forall i,j \in U_{rec}|p_{ij} = 1, \forall ts_i \in TSU \tag{32}$$

– For each course it is not possible to assign more than a specific number of timeslots per day; constraint (14) becomes:

$$\sum_{u \in U_{rec}|f_{cu}=1} \left(\sum_{\substack{ts=g*h+1 \\ ts \in TSR}}^{g*h+h} y'_{ts,u} + \sum_{\substack{ts=g*h+1 \\ ts \in TSU}}^{g*h+h} y_{ts,u} \right) \leq td \quad \forall g \in G, \forall c \in C \tag{33}$$

– For each course it is not possible to assign more than a specific number of timeslots per week; constraint (15) becomes:

$$\sum_{u \in U_{rec} | f_{cu}=1} \left(\sum_{\substack{ts=w*k+1 \\ ts \in TSR}}^{w*k+k} y'_{ts,u} + \sum_{\substack{ts=w*k+1 \\ ts \in TSU}}^{w*k+k} y_{ts,u} \right) \leq tw$$
$$\forall w \in W, \forall c \in C \tag{34}$$

– If in the same day there are two lessons of the same course then they must be consecutive; constraint (16) becomes:

$$\begin{aligned} & y_{ts_1,u_1} + y_{ts_2,u_2} \leq 1 \\ & \forall g \in G, \forall ts_1, ts_2 \in TSU \mid \\ & (ts_2 - ts_1 > 1 \vee ts_2 - ts_1 < -1) \\ & \wedge (ts_1 > g * h \wedge ts_1 \leq g * h + h) \\ & \wedge (ts_2 > g * h \wedge ts_2 \leq g * h + h), \\ & \forall u_1, u_2 \in U_{rec} \mid \exists\, c \in C \wedge f_{cu_1} = 1 \wedge f_{cu_2} = 1 \end{aligned} \tag{35}$$

$$\begin{aligned} & y'_{ts_1,u_1} + y_{ts_2,u_2} \leq 1 \\ & \forall g \in G, \forall ts_1 \in TSR, \forall ts_2 \in TSU | \\ & (ts_2 - ts_1 > 1 \vee ts_2 - ts_1 < -1) \\ & \wedge (ts_1 > g * h \wedge ts_1 \leq g * h + h) \\ & \wedge (ts_2 > g * h \wedge ts_2 \leq g * h + h), \\ & \forall u_1, u_2 \in U_{rec} \mid \exists\, c \in C \wedge f_{cu_1} = 1 \wedge f_{cu_2} = 1 \end{aligned} \tag{36}$$

The sub-MIP objective function is:

$$\begin{aligned} & \min f = \\ & \sum_{u \in U_{rec}} r_u \left(l_u - \left(\sum_{ts \in TSR} y'_{ts,u} + \sum_{ts \in TSU} y_{ts,u} \right) \right) \end{aligned} \tag{37}$$

In Algorithm I we report a pseudo-code describing the steps of our heuristic approach.

V. Computational Results

To assess the quality of our approaches, we have performed a number of experiments on a data set composed by both real-life and randomly-generated instances. Table I reports a description of the instances composition, in terms of:

– whether they are real-life (RL) or randomly-generated (RG);
– number of available teachers ($|D|$);
– number of of courses to recovery ($|C|$);
– number of teaching units to recovery ($|U_{rec}|$);
– number of students involved in the remedial courses ($|S|$).

For each experiment we have imposed a time limit of $3,600$ seconds. To evaluate the performance of the exact and heuristic approaches we have used the percentage objective function gap obtained as:

$$\text{gap} = \frac{(\text{obj}_M - \text{obj}_H)}{\text{obj}_M} 100,$$

Algorithm 1 Pseudo-code of the heuristic procedure

```
Procedure FastHeuristic
  {g_max is the last day of the planning horizon}
  g_max ← max G
  {y' is the part of solution already calculated}
  y' ← [ ]
  {o_new and o_old are used to evaluate the objective function improvements}
  o_new, o_old ← +∞
  {b denotes the amount of funds used}
  b ← 0
  for g = 1 to g_max do
    (y', o_new, b) ← SOLVE(subMIP(g, y'))
    {Solve the sub-MIP problem associated to the time window [1, g]
    considering y' as fixed}
    {If the budget has been completely used}
    if (b = B) then
      break
    end if
    if (o_new < o_old) then
      o_old ← o_new
    else
      if (g > 1 && WEEK(g) ≠ WEEK(g − 1)) then
        break
        {If we have no objective function improvement and we are
        examining a new week we exit from the for loop}
      end if
    end if
    g ← g + 1
  end for
  return y'
end procedure
```

TABLE I
Description of the Test Instances

Instance(s)	RL / RG	$\|D\|$	$\|C\|$	$\|U_{rec}\|$	$\|S\|$
$1-7$	RG	15	25	150	200
$8-16$	RG	15	20	200	200
$17-18$	RL	21	54	500	214
19	RL	14	54	400	210
20	RL	21	54	400	210

where obj_M represents the objective function value obtained by the MIP model, whereas obj_H is the objective function value achieved by the heuristic.

Our experiments have been performed on a Linux machine clocked at 2.67 GHz and equipped with 27 GB of RAM. The MIP and the sub-MIP models have been coded using the mathematical programming language OPL and solved by means of the black-box solver IBM ILOG CPLEX 12.5. Finally, our heuristic has been implemented in Java.

Table II reports the results of our computational campaign, showing for each instance the objective function values obtained by the MIP model and the heuristic (obj_M and obj_H, respectively), as well as the heuristic running time in seconds (T_H) and the percentage objective function gap. We do not report the MIP model running time, because it always reached the time limit before finding the optimal solution. For this reason, it is interesting to report the MIP model percentage optimality gap (Opt. gap).

We observe that the heuristic always converges before the time limit. In particular, in about the 50% of the cases it takes less than half of the available time. In addition, the value of

TABLE II
COMPARISON OF THE RESULTS OBTAINED BY THE MIP MODEL AND THE HEURISTIC

Instance	obj_M	Opt. gap	obj_H	T_H	gap
1	$4,391$	36.48%	$3,601$	$1,431$	17.99%
2	$11,377$	74.14%	$3,871$	$1,415$	65.98%
3	$4,391$	36.51%	$3,752$	$1,776$	14.55%
4	$2,778$	41.79%	$2,033$	$1,407$	26.82%
5	$2,793$	38.94%	$2,192$	$1,405$	21.52%
6	$2,848$	49.31%	$1,980$	$2,005$	30.48%
7	$2,680$	41.28%	$1,950$	$2,002$	27.24%
8	$2,631$	63.01%	$1,386$	$3,345$	47.32%
9	$2,666$	60.20%	$1,345$	$3,327$	49.55%
10	$2,668$	59.78%	$1,365$	$3,332$	48.84%
11	$4,621$	56.86%	$2,300$	$3,363$	50.23%
12	$3,590$	61.47%	$1,838$	$3,355$	48.80%
13	$5,559$	55.44%	$2,841$	$3,342$	48.89%
14	$5,340$	59.80%	$2,870$	$3,359$	46.25%
15	$7,135$	57.95%	$4,102$	$3,355$	42.51%
16	$7,049$	58.34%	$4,124$	$3,334$	41.50%
17	$5,124$	79.53%	$2,627$	$3,286$	48.73%
18	$5,124$	–	$3,353$	$3,070$	34.56%
19	$3,752$	79.07%	$1,925$	$2,599$	48.69%
20	$4,270$	81.61%	$2,401$	$2,622$	43.77%
Average					40.21%

the heuristic objective function is always better than that of the MIP model with gaps depending on the complexity of the instances. In more detail, in the case of instances $1-7$ the average optimality gap is 45.49%, the heuristic converges in less than $2,000$ seconds and the objective function gap varies from a minimum of about 18% to a maximum of about 66%, with an average value of $29,23\%$. In the case of instances $8-16$, the average optimality gap is equal to $59,21\%$, the heuristic uses almost all the time at disposal to converge and the objective function gap ranges between 41.50% and 50.23% with an average value of 47.10%. With respect to instances 17 and 18, the optimality gap is equal to 79.53% (this value has not been reported for instance 18, because it was not possible to find a feasible solution before the time limit), the heuristic converges in about $3,200$ seconds (on the average) and the objective function gap varies from 34.56% to 48.73% with an average gap of 41.65%. In the case of instance 19 the optimality gap is about 79%, the heuristic converges in about $2,600$ seconds and the objective function gap is 48.69%. Finally, for instance 20 the optimality gap is 81.61%, the heuristic converges in about $2,600$ seconds and the objective function gap is 43.77%.

VI. CONCLUSIONS

In this article, we dealt with a variant of the classic course timetabling problem that focuses on the organization of high schools remedial courses. The purpose of such courses is to provide, with limited availability of funds and with a number of teachers at disposal, the best training offer to under-prepared students to succeed in their studies. We have first proposed a MIP model for this problem. Solving this model to optimality with a general purpose solver was possible for small instances only. Thus, in order to deal with real-life-sized instances, we have also presented a heuristic approach. Our tests have shown that the heuristic was able to converge, in about the 50% of the considered instances, in approximately half of the time limit. On the other hand, the MIP model was never solved to optimality within the time limit, with an average optimality gap of about 57%. In addition, the heuristic also obtained an improvement of the objective function value varying from 18% to 66% with respect to the value obtained by the MIP model.

REFERENCES

[1] C. Gotlieb, The construction of class-teacher timetables, in: IFIP congress, Vol. 62, 1963, pp. 73–77.
[2] N. L. Lawrie, An integer linear programming model of a school timetabling problem, The Computer Journal 12 (4) (1969) 307–316.
[3] M. Sørensen, F. H. Dahms, A two-stage decomposition of high school timetabling applied to cases in denmark, Computers & Operations Research 43 (2014) 36–49.
[4] S. M. Al-Yakoob, H. D. Sherali, Mathematical models and algorithms for a high school timetabling problem, Computers & Operations Research 61 (2015) 56–68.
[5] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, Design and implementation of a course scheduling system using tabu search, European Journal of Operational Research 137 (3) (2002) 512–523.
[6] Z. Lü, J.-K. Hao, Adaptive tabu search for course timetabling, European Journal of Operational Research 200 (1) (2010) 235–244.
[7] S. Abdullah, H. Turabieh, On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems, information sciences 191 (2012) 146–168.
[8] A. Gunawan, K. M. Ng, K. L. Poh, A hybridized lagrangian relaxation and simulated annealing method for the course timetabling problem, Computers & Operations Research 39 (12) (2012) 3074–3088.
[9] S. S. Brito, G. H. Fonseca, T. A. Toffolo, H. G. Santos, M. J. Souza, A sa-vns approach for the high school timetabling problem, Electronic Notes in Discrete Mathematics 39 (2012) 169–176.
[10] P. Guo, J.-x. Chen, L. Zhu, The design and implementation of timetable system based on genetic algorithm, in: Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on, IEEE, 2011, pp. 1497–1500.
[11] S. Yang, S. N. Jat, Genetic algorithms with guided and local search strategies for university course timetabling, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41 (1) (2011) 93–106.
[12] C. Nothegger, A. Mayer, A. Chwatal, G. R. Raidl, Solving the post enrolment course timetabling problem by ant colony optimization, Annals of Operations Research 194 (1) (2012) 325–339.
[13] C. Valouxis, E. Housos, Constraint programming approach for school timetabling, Computers & Operations Research 30 (10) (2003) 1555–1572.