Constraint programming approach for school timetabling

Article *in* Computers & Operations Research · September 2003

DOI: 10.1016/S0305-0548(02)00083-7 · Source: dx.doi.org

Efthymios Housos
University of Patras

ARTICLE IN PRESS

PERGAMON

Computers & Operations Research ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

computers & operations research

www.elsevier.com/locate/dsw

Constraint programming approach for school timetabling

Christos Valouxis *, Efthymios Housos

Department of Electrical and Computer Engineering, Computer Systems Laboratory, University of Patras, GR-26500 Rio, Patras, Greece

Received 1 May 2001; received in revised form 1 January 2002

Abstract

In this paper, the timetabling problem for a typical high school environment was modeled and solved using a constraint programming (CP) approach. In addition, operations research (OR) models and local search techniques were also used in order to assist the CP search process by effectively reducing the solution search space. Relaxed models that can be solved using minimum cost matching algorithms were used in order to calculate problem lower bounds at various instances of the solution process. These bounds were in turn used to prioritize the search options of the CP process. The use of minimum cost matching model in the search process is an economical and efficient mechanism for the creation of effective search strategies and it is a competitive manner of introducing problem domain information in the CP environment. By including in the solution process a sequence of local search steps, the solution quality was further improved. Several large problems were solved and actual computational results for specific problem instances are presented.

Scope and purpose

There exist various school timetabling problems depending on the environment and the characteristics of the particular school level [1]. In this paper the high school situation in which the teachers teach in several different class sections during the day and the students remain in their classrooms is modeled and solved. The objective function attempts to minimize the idle hours between the daily teaching responsibilities of all the teachers while also attempting to satisfy their requests for early or late shift assignments. The school timetabling problem is combinatorial and there are several strict organizational and sequence-related rules that must be respected. The problem specifications used in this paper, although they closely describe the situation of a typical Greek high school, are quite general and abstract, which makes the findings of this paper applicable to wider set of school timetabling problems. The specifications mainly focus on the fact that each teacher is scheduled to lecture for a given number of hours at a fixed subset of class sections and the requirement that all the class sections must be always in session without any empty periods in their daily schedules.

* Corresponding author. Tel.: +30-61-997304; fax: +30-61-997316.
E-mail address: valouxis@ee.upatras.gr (C. Valouxis).

PII: S0305-0548(02)00083-7

The integration of constraint programming and operations research techniques for the solution of this problem is one of the main contributions of this paper. The solutions obtained fully utilized the data management and organizational capabilities of the constraint programming approach while being assisted in the search path selection process by techniques and algorithms form the operations research pool of knowledge. The additional information provided by the calculation of efficient lower bounds and the subproblem domain definition and solution strategy presented in this paper, further assists the CP process in selecting promising search paths.

Keywords: Constraint programming; Timetabling; Combinatorial optimization; Minimum cost matching; Local search

1. Introduction

The solution of the school timetabling problem is a cumbersome and time-consuming task often done by hand by specialized teachers of the particular school. The problem is to find a weekly schedule for all the class sections of the school without violating any of the constraints. Since many different teachers do teach in every class section, the main problem constraints forces the solution process to avoid the assignment of teachers to different class sections at a particular period of the day and the opposite. Other significant constraints involve the teachers daily teaching limits, the daily teacher availabilities and the weekly work limits for every teacher. The quality of a timetable as it is defined in this paper depends on the satisfaction of the preferences for early or late shifts by the teachers and the non-existence of idle hours, unless these are planned for non-teaching school activities, during the daily shift of the teachers.

A number of papers in the literature have suggested various algorithmic approaches for the solution of the school timetabling problem. Schaerf in [1] provides an excellent review of the basic timetabling models and solution approaches suggested by the various researchers up to 1999. In [1] it is suggested that there are three main timetabling problem instances (University, School and Examinations) and in accordance to this classification the model of this paper fits very well in the school timetabling class of problems. This is indeed one of the main reasons for the title of this paper. Schmidt and Strohlein [2] and Junginger [3] suggest a heuristic algorithm where the most constrained teachers and class sections are assigned first in order to reduce the computational complexity. Models and algorithms based on integer programming are proposed by Birbas et al. [4], Tripathy [5] and Dimopoulou and Miliotis [6]. Ostermann and de Werra [7] and de Werra [8] suggest an approach where the required timetable is identified by solving a series of network flow problems. Abramson [9] uses a simulated annealing algorithm for the solution of a school timetabling problem and Costa [10] and Schaerf and Schaerf [11] use a tabu search algorithm for the solution of a similar problem. The constraint programming approach that will be used in this paper has been applied to the university, college and high school timetabling by Abdennadher and Marte [12], Stamatopoulos et al. [13], Kang and White [14] and Yoshikawa et al. [15].

The timetabling problem instance we present in this paper attempts to represent the main peculiarities of the Greek high schools although we strongly feel that the algorithmic methodology and the problem solution experiences presented here also hold for other instances of this problem. The solution methodology presented in this paper is facilitated by the use of a constraint programming

(CP) engine. In particular, the ILOG Solver library was used as the primary platform for our implementation and experiments. The CP engine is able to search the whole space in order to identify the best solution but additional search strategies and pruning schemes are necessary for the efficient solution of such combinatorial problems. The large number of relatively tight constraints makes the CP approach quite suitable for this problem. The addition of operations research (OR) techniques for the calculation of lower bounds effectively assists the CP process. The definition of effective sequences of subproblems, based on our previous experience on similar scheduling problems [16], has further improved the solution process. The combination of OR and CP techniques is indeed one of the basic elements of this work.

The following section defines the timetabling problem as it appears in Greek high schools. This section presents a mathematical model of the problem and the main solution methodology. The third and fourth sections describe in more detail the algorithm and the experimental results. The last section draws some basic conclusions based on our computational experience and suggests some future work that could be done in order to improve the interaction between CP and OR techniques.

2. Problem definition and solution process

In Greek high schools, a number of teachers with various educational specializations are available. Every teacher is qualified to teach a certain number of courses. In addition, the students are organized in three school years with every year having a number of class sections. Due to the number of available students per class year, in a typical high school there could be up to four class sections per year. In Greece, the high school years, involve the seventh, eighth and ninth education years for all children of Greece. The lectures that must be given to every class are predefined and common for the whole country. Is should be noted that a teacher can teach a specific lecture to all or some of the class sections and some lectures can be assigned to teachers of various specialties.

The total teaching load of every teacher is different, depending on seniority and other conditions relating to health and/or planned non-teaching activities. A particular teacher can teach in more than one school in order to fulfill his total work requirements. In such a situation, the teacher is available at a particular school only at specific days of the week. The teaching days are Monday–Friday and the daily teaching hours for the students are either six or seven. Every class section has its own permanent room that implies that there is no need for the existence of room constraints in this version of the high school timetabling problem. The teachers are assigned to the class sections before the creation of the timetable and this indeed is one of the main inputs of this model.

The list of the rules considered in our model in order to be able to create an acceptable and viable timetable is

1. Every class section must have in the timetable the specified number of lectures by the proper teachers.
2. Every teacher must have in the timetable the specified lectures for the proper class sections and working days.
3. No teacher can meet two class sections at the same day and hour.
4. Two teachers should not be assigned to the same class section at the same day and hour.

5. The teaching schedule of every class section must be continuous and always starting on the first hour of the day. Any non-teaching period is allowed only in the last hour of the day.
6. We should have a balanced daily workload for all teachers.
7. The weekly teacher presence to the various class sections that he teaches should be balanced among all the days of the week.
8. An effort should be made to respect the teacher preferences for early in the morning or later in the day teaching hours, for all teachers.
9. There must exist a minimum number of non-teaching hours (idle hours) in the middle of a sequence of daily teaching hours.

Rules 1–7 specify the hard legality requirements of the problem and rules 8 and 9 specify the quality characteristics of the timetable to be used in the objective function of the model.

2.1. Parameters and variables

The necessary parameters and data sets needed for the model definition are the following:

- T is the set of teachers.
- C is the set of class sections.
- D is the set of days in the timetable.
- H is the set of teaching hours.
- $TeacherClassHours[t][c]$ is total teaching hours of teacher $t \in T$ for every class section $c \in C$ that must appear in the timetable.
- D_t is the total number of teaching days of teacher $t \in T$.

From the previous parameters and data sets, the following additional sets and parameters are also defined in order to facilitate the model description process:

- T_c is the set of teachers that teach class section $c \in C$.
- C_t is the set of class sections that teach teacher $t \in T$.
- $TeacherTotalHours[t]$ is the total teaching hours of teacher $t \in T$, as the sum of teaching hours $TeacherClassHours[t][c]$ for all class sections $c \in C_t$.
- $ClassTotalHours[c]$ is the total teaching hours of class section $c \in C$, as the sum of teaching hours $TeacherClassHours[t][c]$ for all teachers $t \in T_c$.
- C_L is the set of class sections that do not have a full program for all days. Given that the maximum number of teaching hours for a particular class section is equal to d^*h, where d is the total teaching days and h is the maximum number of teaching hours per day, in the case where a class section c has $ClassTotalHours[c]$ less than this maximum it is implied that this class section belongs in the set C_L.

For the definition of the timetable a number of set variables $ProgrC[c][d][h]$ for every $c \in C$, $d \in D$, $h \in H$ is used. The domain of these variables is the set T_c of the teachers that teach class section c. So when the variable $ProgrC[c][d][h]$ has the value t, means that on day d and hour h teacher t is teaching in class section c.

2.2. Constraints

The values of the variables *ProgrC* must satisfy a number of constraints in order to create a legal timetable. For the definition of these constraints the variables *ProgrC* are used together with some additional variables especially defined for this purpose. These variables were used in order to effectively describe the constraints and the cost function of the model. The symbol #A denotes the cardinality of the set A. The necessary constraints are the following:

In every class section and for each day and hour only one teacher is teaching

$$\#ProgrC[c][d][h] = 1, \quad c \in C,\ d \in D,\ h \in H. \tag{1}$$

In order to model the class sections that finish early, we add

$$\#ProgrC[c][d][|H|] <= 1, \quad c \in C_L,\ d \in D. \tag{2}$$

For the class sections that do not have a full program we have

$$\sum_{d \in D} \#ProgrC[c][d][|H|] = d^{*}h - ClassTotalHours[c], \quad c \in C_L. \tag{3}$$

A teacher cannot be present at more than one class section at the same instance

$$ProgrC[c_1][d][h] \cap ProgrC[c_2][d][h] = \emptyset, \quad c_1, c_2 \in C,\ c_1 \neq c_2,\ d \in D,\ h \in H. \tag{4}$$

The set type variables $ShiftT[t][d]$ where $t \in T$, $d \in D_t$ with domain the set H of the day teaching hours are created in order to express the remaining constraints. When the variable $ShiftT[t][d]$ has the value h, it means that teacher t at day d and at hour h is teaching in some class section. These variables get values whenever some variable of *ProgrC* is assigned values as it is shown in (5)

$$\forall t \in ProgrC[c][d][h] \Rightarrow h \in ShiftT[t][d], \quad t \in T,\ c \in C_t,\ d \in D_t,\ h \in H. \tag{5}$$

Every teacher must teach exactly the pre-specified total number of teaching hours.

$$\sum_{d \in D_t} \#ShiftT[t][d] = TeacherTotalHours[t], \quad t \in T. \tag{6}$$

Constraint (7) is formed in order to achieve a balanced distribution of teaching hours for all working days in accordance with rule 6 of the previous section.

$$\begin{aligned} TeacherTotalHours[t]/|D_t| - 1 <= \#ShiftT[t][d] \\ <= TeacherTotalHours[t]/|D_t| + 1, \quad t \in T,\ d \in D_t. \end{aligned} \tag{7}$$

The teaching hours of every teacher at each class section, define the set type variables $ShiftTC[t][d][c]$ where $t \in T$, $d \in D_t$, $c \in C_t$. When a variable $ShiftTC[t][d][c]$ takes a value h, it means that teacher t at day d and for the particular hour h is teaching at class section c. As it can be seen from (8) these variables are initialized after the definition of the variables *ProgrC* has been done.

$$\forall t \in ProgrC[c][d][h] \Rightarrow h \in ShiftTC[t][d][c], \quad t \in T,\ c \in C_t,\ d \in D_t,\ h \in H. \tag{8}$$

In addition, and according to Eq. (9) these variables *ShiftTC* are used in order to define the work requirement constraint per class section for every teacher.

$$\sum_{d \in D_t} \#ShiftTC[t][d][c] = TeacherClassHours[t][c], \quad t \in T,\ c \in C_t. \tag{9}$$

In order to achieve a class section balanced weekly distribution for the teaching load of every teacher, in accordance with rule 7, Eq. (10) is used.

$$\begin{aligned} &TeacherClassHours[t]/|D_t| - 1 <= \#ShiftTC[t][d][c] \\ &\qquad <= TeacherClassHours[t]/|D_t| + 1, \\ &\qquad t \in T,\ \ d \in D_t,\ \ c \in C_t. \end{aligned} \tag{10}$$

The idle hours of every teacher define the set type variables $SpaceT[t][d]$ where $t \in T$ and $d \in D_t$. The domain of the $SpaceT$ variables is the set H of the daily teaching hours, except for the first and last hours of the day. When the variable $SpaceT[t][d]$ has the value h, it means that teacher t at the day d does not teach at the hth hour and was teaching for a previous and a following hour for this day. As it is shown in (11) these variables get values after the definition of variables $ShiftT[t][d]$ has been done.

$$\begin{aligned} &\forall h \in SpaceT[t][d] \\ &\quad \Leftrightarrow h \notin ShiftT[t][d] \wedge (\exists h' \in \{1 \ldots h-1\} h' \in ShiftT[t][d]) \\ &\quad \wedge (\exists h' \in \{h+1 \ldots |H|\} h' \in ShiftT[t][d])), \quad h \in H - \{1, |H|\}, \quad t \in T,\ \ d \in D_t. \end{aligned} \tag{11}$$

The set of teachers that teach a specific day and hour define the set type variables $ShiftD[d][h]$ where $d \in D$, $h \in H$ with domain the set T of all teachers. When the set variable $ShiftD[d][h]$ includes the value t, it means that teacher t at day d and hour h is teaching in some class section. As it is shown in (12), these variables get values after the definition of $ProgrC$ variables.

$$\forall t \in ProgrC[c][d][h] \Rightarrow t \in ShiftD[d][h], \quad t \in T,\ \ c \in C_t,\ \ d \in D_t, h \in H. \tag{12}$$

The cardinality of these variables is exactly equal to the number of class sections, because all class sections must have a teacher assigned to them. This in quantified in Eqs. (13) and (14). Note that C_L denotes the set of class sections with a reduced daily schedule and (14) exists only if C_L is non-empty.

$$\#ShiftD[d][h] = |C|, \quad d \in D,\ \ h \in H, \tag{13}$$

$$|C| - |C_L| <= \#ShiftD[d][|H|] <= |C|, \quad d \in D. \tag{14}$$

2.3. Definition of the objective function

The objective function (15) of the problem contains two additive components corresponding to the two quality rules presented in the beginning of this section.

$$\mathrm{Min}(cost1 + cost2). \tag{15}$$

The first term of the objective function attempts to model the desirability of the assigned teaching hours. It is calculated using the variables $ShiftT$ of every teacher. For every element of these variables, a penalty is added to *cost1* based on whether the time requests of every teacher are satisfied. For a teacher who prefers early hours the assigned penalties for work during the fifth, sixth and seventh period of some day are significantly larger than the ones of the early hours

$\{0, 2, 5, 10, 20, 50, 100\}$. For a teacher who prefers late hours the penalties are reversed. The *cost1* component is minimized when all teachers are assigned their desired teaching hours.

The second term of the objective function named *cost2*, is the sum of the idle hours for all the teachers in accordance with the definition given to such idle periods earlier in this section. The idle hours were defined by the variables *SpaceT*. For every element of these set variables, a penalty is added to *cost2*. This term is minimized when all teachers have the minimum number of idle hours.

2.4. Lower bound estimation for cost1

A lower bound for the value of *cost1* can be calculated by defining a new problem in which several of the constraints are relaxed. The constraints that are relaxed in this computation are (4), (7) and (10). These constraints refer to the rule that a teacher cannot be present to more than one class section at the same day and hour, the balanced distribution of the total teaching hours for every teacher and the distribution of teaching hours per class section. By eliminating these constraints, the problem is transformed to a minimum cost matching problem. This problem satisfy all the remaining constraints of the model.

The minimum cost matching problem is defined on an undirected graph $G(V, E)$ where V is the set of vertices and E the set of edges. A matching solution set M contains a subset of edges with the property that no two edges of M are connected to a common vertex. If we associate a cost c_{ij} for each edge $(i, j) \in E$, then the minimum cost matching problem is defined as minimize $\sum_{(i,j)\in M} c_{ij}$, where M is a matching solution. For the optimal solution of this problem, the algorithm proposed by Gabow [17] was used.

As it is shown in Fig. 1, which attempts to capture the essence of the lower bound calculation there, are several groups of vertices defined in order to model the specific instance of the problem. The group of vertices on the left side of the graph shown in Fig. 1 contains vertices that correspond to all the teaching hours of all the teachers. In addition, these same vertices are organized in subgroups that correspond to the teaching hours assigned to a particular class section. The sum of these vertices is equal to the total teaching hours of the timetable problem. The second group of vertices contains vertices that correspond to all the teaching hours for all the class sections and for all working days and the third group of vertices contains the set of non-teaching hours for the class sections that do not have a full timetable for every day of the week.

Every vertex of the first group is connected to a vertex of the second group, if the teacher that corresponds to the first vertex can teach the class section at the hour and day that corresponds to the second vertex. These connections are defined only for the same class sections of the first and the second group of vertices. The cost of such an edge is the penalty defined for this assignment. Every vertex of the third group is connected to a vertex of the second group that corresponds to the last teaching hour for the set of class sections that do not have a full timetable. The cost of these edges are a negative number, so the solution of the minimum cost matching problem includes the proper set of these edges satisfying the rule that the non-teaching hours of class sections are at the end of a working day.

The solution of the minimum cost-matching problem connects every vertex of the second group to only one vertex of either the first or the third group. It can be easily shown that if there is a non-connected vertex of the first or the second group, then the given timetable problem is infeasible. This can be better visualized by realizing that an isolated vertex of the first group means that will

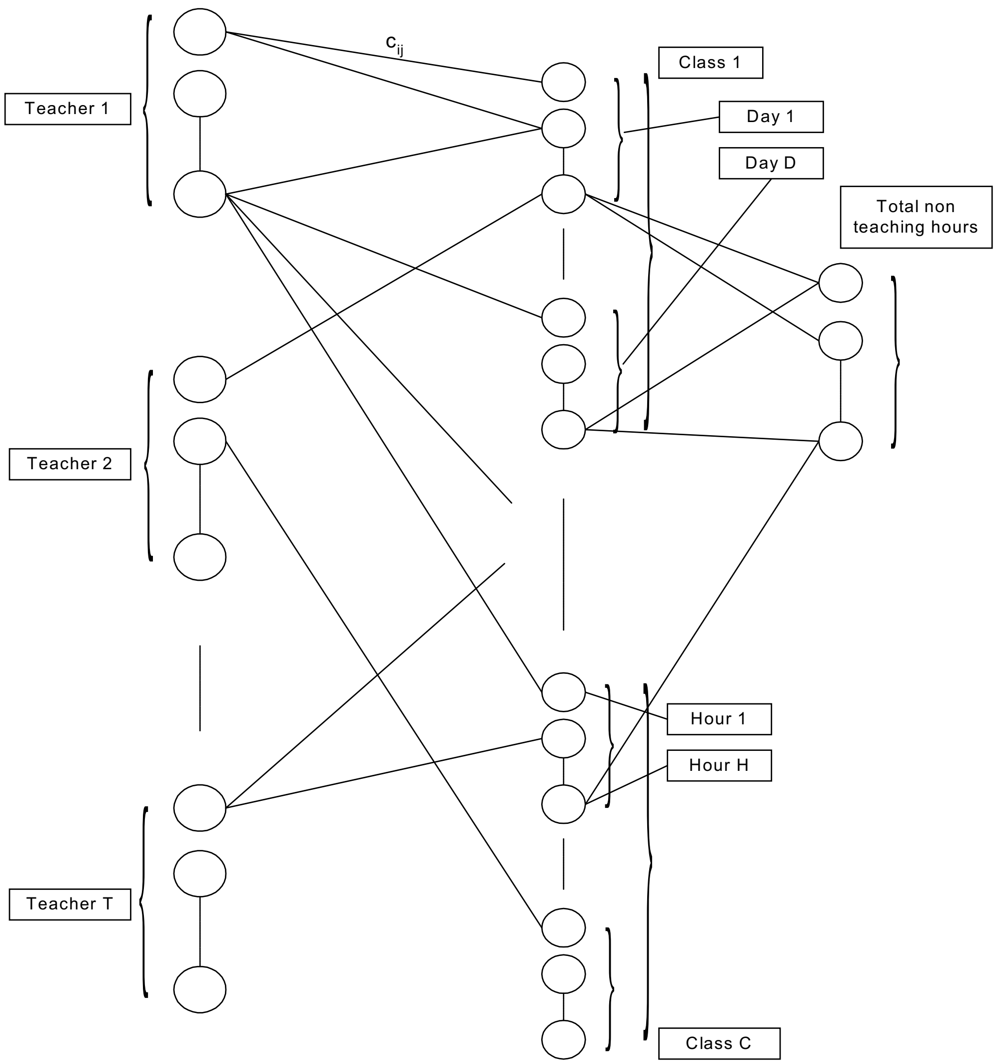

Fig. 1. The graph for the calculation of the lower bound of the *cost1*.

not be possible for the particular teacher to ever satisfy his weekly requirements. This could happen if indeed the total weekly teaching hours for the teachers are not equal to the total weekly teaching load for all class sections, as the problem definition requires. Following a similar logical path an isolated vertex of the second group means that a class section could not ever find a teacher for all of its teaching needs because obviously the sum of the hours all the teachers come to this class section is not the same as the total teaching hours of this class section. The cost of the minimum matching solution without the negative costs corresponding to connections of the third group is an appropriate lower bound of *cost1*. The main mathematical rational for this lower bound calculation, stems from the fact that the addition of more constraints to a minimization problem can only increase or in the best case not affect the optimal value of the problem.

2.5. Constraint programming (CP) for the solution of the timetabling problem

The previous timetabling problem was defined as a constraint satisfaction problem (CSP), consisting of a set of variables and a set of constraints. For each variable, a finite set of possible input values is defined and the constraints main role is to restrict the values that the problem variables can simultaneously take. A solution to a CSP is an assignment of a value from its domain to every variable, in such a way that every constraint is satisfied. In the particular case of this paper, it is required to find an optimal or if the process runs out of time a reasonable solution given the objective function defined in the previous section. There are several products whose task is to facilitate the programming and the modeling for such a problem using special add-on modules to traditional languages like Prolog, C and C++. The ILOG solver library was used in this experiment.

The solutions of the problem are found by searching systematically through all the possible assignments of values to variables. The most important aspect of the solution process when the constraint-programming framework is used, involves the strategy for choosing both the variables to be assigned values and the values assigned to them. In our model, a timetable solution is defined when values have been assigned to all the variables *ProgrC*. These values must satisfy all the constraints while attempting to minimize the cost function. Instead of assigning values to the variables *ProgrC* and in order to better control the search process we first assign values to the variables *ShiftD* that define the set of teachers, which teach at all the available class sections at a specific day and hour. Based on the problem model and the values assigned to the *ShiftD* variables it is possible to assign values to the *ProgrC* variables.

This solution procedure was used because the *ShiftD* variables by being directly connected with the cost function are easier to gage and assign proper values to them. This in addition allows for an efficient search strategy that avoids the examination of solutions with similar costs. Thus, manages to move to better solution neighborhoods faster without being trapped in relatively flat solution areas.

In order to choose a variable $ShiftD[d][h]$ we must firstly select a day and a specific hour within the day. For every working day we start with the variables of the first teaching hour and then with the variables of the last teaching hour. Given the preferences of teachers for early or late teaching assignments, we can easily define values for these variables. In addition, do note that when a teacher has been chosen to start teaching at the first hour of a day it is very rare or often impossible to continue teaching until the last hour. Then we choose variables for the second teaching hour and for the hour before the last hour of the day. Complementary, when a teacher has been chosen to start teaching at the first or the last hour it is most probable and often desired to continue teaching for at least one more hour because the daily work load of a teacher in most situations includes three to four teaching hours per day. This is sequentially done for all the teachers available.

The values assigned to the *ShiftD* variables are selected in a locally optimal manner with the help of a minimum cost matching model. The graph of the matching model has two main groups of vertices. The first group contains vertices that correspond to all the teachers in the domain of variable *ShiftD* and the second group contains vertices that correspond to all the class sections that require a teacher at the particular hour. Every vertex of the first group is connected to a vertex of the second group if the generated connection is valid. This means that the particular teacher can teach at the particular class section that the generated link implies. In mathematical terms this can only be legal if the teacher in question is contained in the domain of the variable *ProgrC* of the particular class section. The cost of such a link depends on the penalty value of such an assignment. The actual penalty cost used at a particular connection decreases if some or all of the following situations do hold.

1. Variables *ProgrC* and/or *ShiftD* received a value corresponding to a particular teacher because of some constraint propagation of previous assignments of the problem variables.
2. A teacher who was active at some previous hour, remains active for some future hour of the same day, and has not yet reached the maximum number of hours per day.
3. A teacher who was teaching either at the previous or at the next hour, from the hour presently examined, and has not yet reached the minimum number of hours per day.

The first case receives a very low penalty cost because it must be always selected in order to achieve the creation of a feasible solution. Cases 2 and 3 are treated in a special manner because they bias the process in a manner that minimizes the idle hours for the teachers. The solution of the minimum cost matching problem guarantees that for all chosen teachers there exists a feasible assignment to actual class sections which means that no two teachers are meeting the same class section at the same time and vice versa. The teachers proposed by the matching solution are sorted in ascending order and are utilized in the assignment of values to the variables *ShiftD*.

After the definition of a single $ShiftD[d][h]$ variable follows the definition of all the variables *ProgrC* that are affected by the particular day d and hour h values. Given that there exist many options at this point of the search, a legal assignment regarding a subset of the problem constraints whose task is to attempt to drive the process towards a balanced work schedule for all teachers is desired. This also helps to indirectly satisfy the strategic goal of attempting to delay the full utilization of the teaching resources as long as possible.

Following the previous solution process, all variables *ProgrC* would have been assigned values and an initial timetable would have been constructed. Afterwards, the CP framework is allowed to further search for better solutions by adding a new artificial constraint to the problem that basically forces the CP system to suggest solutions with improved cost at all times. This constraint is treated in a manner similar to all the other constraints and the constraint propagation system infrastructure automatically excludes all of the solution space that contains inferior solutions. Considering the fact, that this constraint needs the definition of a large number of problem variables in order to be activated and that the total search space in real problem situations is extremely large, additional search strategies need to be introduced in order to improve the search and reduce the time requirements.

2.6. Space reduction search strategies

After the first solution of the timetable problem has been located, a better solution for the particular objective function can be found either if the total number of idle hours is decreased or an assignment to more desired teaching hours for some or all of the teachers is achieved. For efficiency purposes the search is discontinued if one or more of the following conditions occur:

- If in the solution just identified the total number of idle hours (*cost2*) is zero and the term *cost1* is equal to the lower bound of *cost1*. This implies that the optimal solution has been found.
- If the best solution up to this point does contain idle hours and the term *cost1* is equal or very close to the lower bound of *cost1*, the search is allowed to proceed only to areas where the total idle hours of the current solution are reduced beyond the ones of the best solution.
- When during the search the total idle hours of the current solution are greater than the idle hours of the best solution, the search stops investigating this area of solutions, because a solution with less idle hours is always more desirable.
- When during the search the total idle hours of the current solution is equal to the idle hours of the best solution, the search does continue for this area of solutions only if *cost1* can be improved in the process. Again a lower bound of *cost1* can be calculated in a manner similar to the one described in Section 2.4. For this matching problem, all the previous variable definitions must be respected and the formulated graph thus becomes significantly smaller. The cost of the solution of the minimum cost matching problem is the best value that can be found for this solution area. If this cost is not less than the equivalent cost of the best solution, the search for this area of solutions is abandoned.

2.7. Local search for 1 and 2 days

In this section, an efficient search mechanism that is used in order to further improve the solution quality, after the CP-based procedure has finished due to time to solution limitations, is presented. The main idea involves the freezing of complete days of the solutions and the re-examination only of single days in order to better examine all the local combinations. This could be also done for two intervals of two free days and three frozen days as well. Sequential improvements for 1 or 2 days are significant and indeed possible due to the additive and independent nature of the work performed by the teachers on every day of the week.

The constraints that are introduced in order to enforce a local search for a single day are such as to disallow changes of the values of the problem variables that correspond to the all the remaining days. In the local subproblem constraints that guarantee the legality of the solution, when variables of this day are changed, and the remaining days are frozen are also included. The new constraints guarantee that the total teaching hours of every teacher remain the same as in the entire initial solution and these particular teaching hours remain assigned to the same class sections in order not to violate the balanced solution quality and remain feasible. The subproblem search strategies remain the same as before. This local search process is repeated for all working days. This is also done for some of the larger problems using a 2-days search space for the local search.

3. The complete algorithm for the solution of the school timetabling problem

Based on the various tools and processes presented in the previous section the complete algorithm used for the solution of the school timetabling problem is shown in Fig. 2.

```
Repeat
      Solve the entire problem (section 2.5)
      Repeat
            For every day
                  Solve the local problem (section 2.7)
            For every 2-days
                  Solve the local problem (section 2.7)
      Until there are no more improvements
Until the stopping criterion
```

Fig. 2. The algorithm for the solution of the timetabling problem.

In all steps of the solution process the searching performed by the CP engine is assisted by various computational steps whose task is to reduce the search space while emphasizing the most promising paths of the search tree. According to our experiments, by utilizing our algorithmic steps and specific problem domain information, the CP engine achieves significantly better solutions for a fixed computation time than a CP search that only includes standard library functions and search discrimination strategies.

The solution process presented in Fig. 2 starts by solving the complete timetable problem for the whole week. The execution time of this step whose task is to find a solution for the entire problem is limited to 10 min or until a feasible solution has been found. Then several local search strategies take effect and attempt to improve the solution. The 1- and 2-days local search process is performed on all possible combinations. The maximum execution time for the local search procedures is limited to 2 min per sub problem. The local search step is skipped whenever the current solution is locally optimal. The solution is locally optimal if the total idle hours for 1 or 2 days are zero and the term of the objective function that corresponds to the time desirability constraints (*cost1*) is equal to its lower bound.

Every improved solution is saved at the output queue of the process, and the algorithm keeps searching for additional solutions whose cost is superior to the one already saved. If the 2-days local search improves the current solution, then the single-day local search is also executed for the days involved in the 2-days sub problem. Do note that for all major iterations of the algorithm of Fig. 2, the 1- and 2-days local search procedures are not executed, if the timetable for the corresponding days has not been changed.

The overall stopping rule of the algorithm is the total execution time and for these experiments has been defined to be 1 h. HP 9000/715/100 MHz workstations have been used for these experiments. Of course, the algorithm is terminated earlier if the optimal solution has been found before the maximum allowed computation time has been reached.

4. Computational results

The complete algorithm presented in the previous sections has been implemented using C/C++ and for the CP framework, the C/C++ version of the ILOG Solver was used [18]. In more detail, the ILOG Solver is a C++ library that includes all the necessary data types and functions that are required to express problem constraints and define various search strategies and bounds. The algorithm was applied for the solution of small, medium and large school timetabling problems. The data of a typical school timetabling problem is presented in Table 1. The first column of the table contains a list of the available teachers, each having a different identification code (id). In the given problem, there exist six class sections. The required teaching hours per week for every teacher ("total hours" column) and their breakdown to the various class sections is presented in the second set of columns. The "days" column presents the weekdays that each teacher is available at the particular school given that some teachers do teach in more that one school. The last column shows the teacher desirability for early or late daily shifts. The last row of the table contains the total weekly teaching hours for each class section. Do note that all class sections meet from Monday to Friday only.

The timetable produced using the solution process of this paper is presented in Tables 2 and 3. Table 2 contains the weekly timetable for each class section and Table 3 contains an equivalent view of the timetable for use by the teachers. The teachers view is important in order to better visualize the presence of idle hours and clearly evaluate if the requested shift preferences have been achieved. In this particular problem, the obtained solution is optimal because the value of the *cost1* component is equal to the calculated lower bound of *cost1* and there exist no idle hours in the solution. Besides this test problem, the algorithm was also applied to a set of medium to large timetable problems

Table 1
The data of the timetable problem

Teacher	Class sections						Total hours	Days					Desired shift	
Id	C1	C2	C3	C4	C5	C6		Mo	Tu	We	Th	Fr	Early	late
T1	3	3	3	3	2	2	16	√	√	√	√	√		•
T2	9		8				17	√	√	√	√	√	•	
T3		9		8			17	√	√	√	√	√	•	
T4	2				8	8	18	√	√	√	√	√	•	
T5		2	2	2	4	4	14	√	√	√	√			•
T6	2	2	2	2	2	2	12	√	√	√				•
T7	2	2	2	2	2	2	12	√	√	√	√	√		•
T8	3	3	3	3	2	2	16	√	√	√	√	√		•
T9	1	1	1	1	1	1	6				√	√		•
T10			1	1	1	1	4					√	•	
T11	4	4	4	2	2	2	18	√	√	√	√	√		•
T12	4	4	4	4			16	√	√	√	√	√	•	
T13				2	6	6	14	√	√	√	√		•	
T14	3	3	3	3	2	2	16	√	√	√	√	√		•
T15	2	2	1	1			6	√				√		•
Total	35	35	34	34	32	32	202							

Table 2
The timetable for all class sections

Class section	Day	Hour						
		H1	H2	H3	H4	H5	H6	H7
C1	Monday	T2	T8	T12	T2	T14	T15	T1
	Tuesday	T2	T12	T4	T2	T6	T1	T11
	Wednesday	T2	T2	T12	T1	T6	T11	T8
	Thursday	T9	T12	T2	T4	T14	T7	T11
	Friday	T2	T2	T7	T8	T14	T11	T15
C2	Monday	T3	T12	T8	T11	T15	T5	T6
	Tuesday	T5	T3	T3	T12	T11	T14	T7
	Wednesday	T3	T12	T3	T6	T11	T1	T8
	Thursday	T3	T3	T9	T14	T8	T1	T7
	Friday	T3	T3	T12	T11	T15	T8	T1
C3	Monday	T3	T12	T8	T11	T15	T5	T6
	Tuesday	T5	T3	T3	T12	T11	T14	T7
	Wednesday	T3	T12	T3	T6	T11	T1	T8
	Thursday	T3	T3	T9	T14	T8	T1	T7
	Friday	T3	T3	T12	T11	T15	T8	T1
C4	Monday	T12	T3	T3	T8	T11	T1	T14
	Tuesday	T3	T13	T12	T3	T14	T8	—
	Wednesday	T5	T3	T13	T3	T1	T14	T6
	Thursday	T12	T9	T3	T1	T7	T11	T5
	Friday	T7	T12	T3	T10	T6	T15	T8
C5	Monday	T13	T13	T4	T4	T7	T14	T5
	Tuesday	T4	T4	T5	T11	T1	T6	T14
	Wednesday	T13	T13	T4	T5	T8	T9	—
	Thursday	T13	T13	T4	T8	T11	T5	—
	Friday	T4	T4	T10	T7	T1	T6	—
C6	Monday	T4	T4	T13	T13	T5	T6	T11
	Tuesday	T13	T5	T13	T4	T8	T7	T6
	Wednesday	T4	T4	T5	T13	T9	T8	—
	Thursday	T4	T4	T13	T5	T1	T14	—
	Friday	T10	T7	T4	T14	T11	T1	—

of typical Greek high schools. The data of these problems and the results obtained are shown in Table 4. The first column of Table 4 refers to the number of the problem. The second and the third columns show the number of available class sections and the number of teachers, respectively. The fourth column contains the total teaching hours for all lectures and for all class sections of the particular school and it is shown as a problem size unit. The number of teaching hours per day is six or seven and the number of working days is equal to five for all problems. The fifth column

Table 3
The timetable for all teachers

Days																																				
Teacher	Monday							Tuesday							Wednesday							Thursday							Friday							Total
	H1	H2	H3	H4	H5	H6	H7	H1	H2	H3	H4	H5	H6	H7	H1	H2	H3	H4	H5	H6	H7	H1	H2	H3	H4	H5	H6	H7	H1	H2	H3	H4	H5	H6	H7	
T1						C4	C1				C3	C5	C1					C1	C4	C2	C3				C4	C6	C2					C3	C5	C6	C2	16
T2	C1	C3	C3	C1				C1	C3	C3	C1				C1	C1	C3					C3	C3	C1					C1	C1	C3					17
T3	C2	C4	C4					C4	C2	C2	C4				C2	C4	C2	C4				C2	C2	C4					C2	C2	C4					17
T4	C6	C5	C6	C5				C5	C5	C1	C6				C6	C6	C5					C6	C6	C5	C1				C5	C5	C6					18
T5					C6	C2	C5	C2	C6	C5					C4	C3	C6	C5							C6	C3	C5	C4	—	—	—	—	—	—	—	14
T6					C3	C6	C2					C1	C5	C6				C2	C1	C3	C4	—	—	—	—	—	—	—					C4	C5		12
T7				C3	C5							C3	C6	C2	—	—	—	—	—	—	—-					C4	C1	C2	C4	C6	C1	C5				12
T8	C3	C1	C2	C4								C6	C4						C5	C6	C1				C5	C2	C3					C1	C3	C2	C4	16
T9	—	—	—	—	—	—	—	—	—	—	—	—	—	—					C6	C5		C1	C4	C2	C3				—	—	—	—	—	—	—	6
T10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	C6	C3	C5	C4				4
T11				C2	C4	C3	C6				C5	C2	C3	C1				C3	C2	C1						C5	C4	C1				C2	C6	C1	C3	18
T12	C4	C2	C1					C3	C1	C4	C2				C3	C2	C1					C4	C1	C3					C3	C4	C2					16
T13	C5	C6	C5	C6				C6	C4	C6					C5	C5	C4	C6				C5	C5	C6					—	—	—	—	—	—	—	14
T14					C1	C5	C4					C4	C2	C5					C3	C4	C2				C2	C1	C6	C3				C6	C1	C3		16
T15					C2	C1	C3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—					C2	C4	C1	6
Total	6	6	6	6	6	6	6	6	6	6	6	6	6	4	6	6	6	6	6	6	4	6	6	6	6	6	6	4	6	6	6	6	6	6	4	202

Table 4
Data and results of various timetable problems

No. of problem	No. of class sections	No. of teachers	No. of teaching hours	*Cost1*—lower bound	Solution	
					Cost1	No. of idle hours
1	5	11	150	1415	1415	0
2	6	15	202	1535	1535	0
3	7	17	210	960	990	0
4	9	23	315	1323	1395	1

contains the theoretically calculated lower bound of *cost1*, which is a metric of the mismatches between the desired and the actual daily shift for all teachers. The last columns present the actual values of *cost1* in the produced timetable and the total number of idle hours for all teachers. For the first and second problems, the algorithm indeed finds the optimal solution. The execution time for these two problems is from 15 to 20 min. For the other two larger problems, the solutions are also very satisfactory. The value of *cost1* is very close to its absolute lower bound while the idle hours are zero for the third problem and just one for the fourth problem. The execution time for both of these problems is 1 h meaning that the maximum allowed computation time has been utilized. Do note that for the last two problems, a CP process that does not utilize the local search strategies and the bounds and pruning assistance presented in this paper, is unable to find satisfactory solutions in an acceptable computation time. For the smaller problems, a solution was found but was quite far from the optimum solution and was only partially acceptable by the teachers.

5. Conclusions

In this paper, an algorithm based on CP for the solution of the school timetabling problem is presented. The overall problem is solved in acceptable time and the objective value of the solutions obtained is quite close to their optimal values. The suggested algorithm takes advantage of the properties of the set type variables in order to effectively define the constraints and the objective function of the problem. The timetabling problem is combinatorial in nature because there exist many teachers to be scheduled and for every teacher a very large number of different alternatives exist. The quality of a timetable depends on the ability to assign the teaching hours of each teacher to their most desirable periods of the day and in accordance with their requested preferences. The minimization of the idle hours for the daily work patterns of each teacher is also important. A lower bound for the early or late shift request component of the cost function is computed solving a minimum cost matching problem defined by relaxing some of the constraints of the complete model. This bound is used mainly for the estimation of the solution quality of each possible path and can be used in order to prove the optimality property of the final solution. Local search schemes for 1- and 2-days sub problems are also embedded in the algorithm and assist the solution process in finding the best possible solution for a given computation time interval.

The suggested solution approach could be also used for solving various other timetabling problems. The set type variables can effectively express the timetable constraint and only the search strategies must be adapted to the characteristics of the objective function. The essence of the customization for different problem situations has to do with the fact that the search engine will possibly investigate many non-fertile solution areas unless properly guided. In the early stages of this work and without the continuous use of minimum cost matching for the efficient implementation of the search strategy the process would either find an unacceptable solution or return without any solution for the given time execution window. The investigation and creation of polynomial time optimization models like the minimum cost matching algorithm in order to reduce the search space is extremely important for the efficiency of CP-based scheduling systems.

The column generation technique for the solution of timetabling problems should also be investigated given its recent success in several other scheduling domains [19]. The ability to dynamically find the correct missing variable for the current sub-problem appears extremely interesting and powerful. The comparison of constraint programming approaches with more traditional operations research-based approaches and the possible connection of the two schools of thought appears as a very promising and interesting area of research.

References

[1] Schaerf A. A survey of automated timetabling. Artificial Intelligence Review 1999;13:87–127.
[2] Schmidt G, Strohlein T. Timetable construction—an annotated bibliography. The Computer Journal 1979;23:307–16.
[3] Junginger W. Timetabling in Germany. Interfaces 1986;16:66–74.
[4] Birbas T, Daskalaki S, Housos E. Timetabling for Greek high schools. Journal of the Operational Research Society 1997;48:1191–200.
[5] Tripathy A. Computerized decision aid for timetabling—a case analysis. Discrete Applied Mathematics 1992;35: 313–23.
[6] Dimopoulou M, Miliotis P. Implementation of a university course and examination timetabling system. European Journal of Operational Research 2001;130:202–13.
[7] Ostermann R, de Werra D. Some experiments with a timetabling system. OR Spectrum 1983;3:199–204.
[8] de Werra D. An introduction to timetabling. European Journal of Operational Research 1985;19:151–62.
[9] Abramson D. Constructing school timetables using simulated annealing: sequential and parallel algorithms Management Science 1991;37:98–113.
[10] Costa D. A tabu search algorithm for computing an operational timetable. European Journal of Operational Research 1994;76:98–110.
[11] Schaerf A, Schaerf M. Local search techniques for high school timetabling. Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling, 1995. p. 313–23.
[12] Abdennadher S, Marte M. University course timetabling using constraint handling rules. Journal of Applied Artificial Intelligence 2000;14(4):311–26.
[13] Stamatopoulos P, Viglas E, Karaboyas S. Nearly optimum timetable construction through CLP and intelligent search. International Journal on Artificial Intelligence Tools 1998;7(4):415–42.
[14] Kang L, White M. A logic approach to the resolution of constraints in timetabling. European Journal of Operational Research 1992;61:306–17.
[15] Yoshikawa M, Kaneko K, Nomura Y, Watanabe M. A constrained based approach to high school timetabling problems: a case study. Proceedings of the 12th National Conference On Artificial Intelligence (AAAI-94), 1994. p. 1111–6.
[16] Housos E, Elmroth T. Automatic subproblem optimization for airline crew scheduling. Interfaces 1997;27(5):68–77.
[17] Gabow H. An efficient implementation of Edmonds algorithm for maximum matchings on graphs. Journal of ACM 1975;23:221–34.

[18] ILOG. ILOG Solver 4.4 Reference Manual, 1999.
[19] Valouxis C, Housos E. Combined bus and driver scheduling. Computers and Operations Research 2002;29:243–59.

Christos Valouxis earned his Ph.D. degree in the Department of Electrical and Computer Engineering of the University of Patras. He has published several articles in the area of scheduling and timetabling. His main research interests are algorithm engineering, combinatorial optimization, column generation and the interaction between OR and CP approaches.

Efthymios Housos is a professor and head of the Computer Systems Laboratory of the Department of Electrical and Computer Engineering at the University of Patras. He earned his BS, MS and Ph.D. degrees at Columbia University in New York. He worked for several years in the Advanced Decision Support Systems of AT& T Bell Laboratories and has published several papers in the area of scheduling, optimization and other efficient solutions to hard combinatorial problems.