Journal of Computer Science (Special Issue) : 64-71, 2005
ISSN 1549-3636

Modelling the Timetabling Problem Using Goal Programming

[1]Jihad Mohamad Jaam, [2]Mohamed Larbi Rebaiaia and [1]Ahmad Mojahid Hasnah
[1]Computer Sciences Department, University of Qatar, Qatar
[2]Computer Sciences Department, University of Batna, Algeria

Abstract: The school timetabling problem is a well known combinatorial hard problem. It consists mainly of allocating timeslots to lectures from a limited number of available timeslots taking into consideration the number of teachers and classrooms. Many other soft constraints may also be present, like teachers wishes, distances between different classrooms, break between lectures, classrooms locations, etc. However, a solution can be accepted without satisfying all the soft constraints and an optimal solution is obtained whenever all the pre-determined constraints are satisfied. Many heuristic algorithms and modelling approaches have been proposed to solve the timetabling problem. However, they deal with particular instances of the problem and no general solution can be found for all instances. In this research we propose a new multi-objective model for the timetabling problem using goal programming. We show that our model is very flexible and many other soft constraints related to the timetabling problem can be added easily to the model. Our experiments in solving the proposed model show that the obtained results are very promising.

Key words: Scheduling, timetabling problems, multi-objectives

INTRODUCTION

The school timetabling problem is a scheduling problem which consists of allocating timeslots to a combination of students, teachers, lectures and classrooms under different types of hard and soft constraints, within a limited number of permitted timeslots, in such a way that none of the specified constraints are violated. Frequently, the problem could be highly constrained due to the wide range of models in different educational entities. In fact, many soft constraints may be present, like teachers wishes, distances between different classrooms, break between lectures, classrooms locations, etc. The non-satisfaction of some of these soft constraints can be accepted in some circumstances. However, an optimal solution consists of satisfying all the pre-determined constraints.

It is well known that the school timetabling problem is a combinatorial hard problem and many techniques have been developed recently to solve the problem and the solutions are obtained according to each particular instance of the problem, as no general solution can be found for all instances due to the exponential nature of the problem. These techniques include constraints programming[1-3], neural network[4], heuristics algorithms[5], genetic programming[2,6,7], simulated annealing[8], Tabu search[9,10], integer programming and branch and bound[9]. Other interesting research works on the timetabling problem are published also in[1,11-13]. In addition, a popular international bi-annual conference called PATAT is dedicated mainly to the timetabling problem.

A given model for the timetabling problem is a mixture of objective functions with a set of soft and hard constraints. The objective functions are used to model the desire and the aspiration of the decision makers in order to assign lectures to teachers taking into consideration some constraints like the precedence constraints, the classrooms capacities, the teacher wishes, the distances between classrooms, etc. Some modelling approaches associated with the timetabling problem are given by Elkhiary[11], Elkhiary *et al.*[12], Abdennadher *et al.*[1], Burke *et al.*[6], Schaerf[14] and Thomson *et al.*[8]. Multiple criteria approaches for the timetabling problem have been also discussed in[15-17].

In this research, we propose a new multi-objective model for the timetabling problem using goal programming where a timetable has to offer the teachers and the student's maximum free times between two consecutive lectures and optimise the dead time of the classrooms occupancies.

GOAL PROGRAMMING

The goal programming technique has been introduced first by Charnes, *et al.*[18] and extended later by Charnes and Cooper[19]. It deals mainly with the problem of achieving a set of conflicting goals. Two types of goals are commonly used in goal programming: a two-sided goal in which any deviation, either upward or downward, would result in some penalty (exact achievement); and a one-sided goal in which only upward or downward deviation would be penalized. Fuzzy methodologies are used whenever uncertainties exist in the treated problem[20].

Corresponding Author : Jihad Mohamad Jaam, Computer Sciences Department, University of Qatar, Qatar

There are many applications of goal programming that include water management systems[21-23], location problems resolution[24] and decision making problems[25]. Some good literatures on theory and applications of the goal programming can be found in[25-27].

In linear programming problems there is a single objective function to be maximised or minimised (subject to a set of constraints). In some problems there may be more than one objective (or goal) and we need to trade-off objectives against each other.

One approach to handle with problems of multiple objectives is to choose one of the goals as the master goal and to treat the others as classical constraints to ensure that some minimal "satisfaction" level of the other goals is achieved. However, goal programming provides more satisfactory solutions where problems can be solved using standard linear programming algorithms.

In goal programming a problem means some particular instance of the following model: a polyhedron $K \subseteq \Re^{n}$ is given as the set of decisions; there exist *m* criteria matrices denoted by, $C_1, \cdots, C_m$, with C_j in $\Re^{n \times n_j}$; each decision $x \in K$ is valued according to a given criterion C_j with the vector $C_j^T x$, to be compared with a given target set $T_j \subseteq \Re^{n_j}$. The deviation $d_j(x)$ of the decision *x* with respect to the target set T_j is defined as $d_j(x) = \inf_{z_j \in T_j} \gamma_j \left(C_j^T x - z_j \right)$ for some given γ_j, while the overall deviation at *x* is measured by $\gamma\left(d_1(x), \cdots, d_m(x)\right)$ and where γ is a norm in the polyhedron $\Re^{m}$ assumed to be monotonic in the nonnegative set $\Re_{+}^{m}$.

Example: Assume a decision maker would like to maximize the profit of some products. Let x_1 and x_2 be the decision variables representing such products and let $F(x_1, x_2) = 5x_1 + 7x_2$ be the economic function. Assume that b is the aspiration level and the decision maker wants to make a profit of at least \$500 during a given period. Thus b will be set to 500 ($b = 500$) and we have the following constraint : $5x_1 + 7x_2 \geq 500$. Suppose that $\bar{d}$ and $\overset{+}{d}$ are two deviation variables used to model, over and under achievements respectively. By using goal programming the initial problem, denoted by (A) will be replaced by another problem, denoted by (B) and the objective function acts as a classical constraint.

$$Maximize\ F(x_1, x_2) = 5x_1 + 7x_2 \qquad \text{(A)}$$

$$\begin{cases} Minimize & (\bar{d} + \overset{+}{d}) \\ subject\ to & 5x_1 + 7x_2 + \bar{d} - \overset{+}{d} = 500 \end{cases} \qquad \text{(B)}$$

Obviously, the economic function cannot represent fully the problem. It should have a set of other constraints. Thus the function in (A) and the system in (B) are systematically extended with other constraints. The solution of the problem may be obtained by using an optimisation algorithm, like the Branch and Bound or the heuristic techniques.

PROBLEM MODELLING

We propose a generic model for the weekly lectures timetabling problem. The model is composed by a set of resources (teachers, classes, classrooms, etc), a set of activities (lectures, courses, sport activities, etc) and a set of constraints between these activities. The time is divided into time units or timeslots with a unique duration approach. We consider two types of timeslots: one hour type for ordinary courses and two hours type for sport activities and assimilated. We will have to deal with the time preferences as resource aspirations, which are treated as soft constraints. At the same time, we have to respect some forbidden activities and treat them as hard constraints. We describe an activity by its duration, its time preferences and by a set of resources assigned to it. Similarly, we define a resource as a physical mean described by time preferences and restricted to be assigned to just one activity. The relationships between resources, activities and preferences are given using a common mathematical formula *X (i, j, k, l, s, m, n)*, which means that the class *(i, j)* is assigned to the course *k* with the teacher *l* using the classroom *s*, during the n^{th} timeslot of the day *m*. For instance, *X (2, 3, 1, 2, 9, 5, 3)* represents the section (class or group): second group of the third year is assigned to the course numbered by one (e.g. Maths) with the teacher two (e.g. Mr R. Roy) using the classroom number 9, during the third timeslot (i.e., from 10:00h to 11:00h) of the day Friday. The solution of the problem must satisfy all the hard constraints given in the following relations:

Given a set of levels (years) $Y = \{i_1, i_2, \cdots, i_{|\pi|}\}$, a set of teachers $T = \{l_1, l_2, \cdots, l_{|\varepsilon|}\}$, a set of sections (groups or classes) $J = \{j_1, j_2, \cdots, j_{|\chi|}\}$, a set of courses $K = \{k_1, k_2, \cdots, k_{|\mu|}\}$, a set of rooms, classrooms and laboratories $S = \{s_1, s_2, \cdots, s_{|\rho|}\}$, a set of timeslots

$N=\{n_1,n_2,\cdots,n_{|\eta|}\}$ (slots per day) and $M=\{m_1,m_2,\cdots,m_6\}$ the days of the week, the goal of courses timetabling is to obtain an assignment where each course is assured by a teacher l of the course k to a group j of the year i (i.e., level i) in a classroom s during a timeslot n taken in the day m of the week. The result of such an assignment is a timetable represented by a set of ordered 7-tuple *(i, j, k, l, s, m, n)*. A timetable is called *feasible* if it satisfies all required constraints. Otherwise, it is identified as *unfeasible*. A fundamental requirement in course timetabling is to prohibit time *holes*, called also dead time in a day, which corresponds to the non use of a classroom (or a group of students). The major problem appears in the forbidden assignment with some conflicts. For instance, a student has to take two or more courses in the same timeslot, or a teacher is assigned to two or more groups in the same timeslot, or a teacher has to teach three or more consecutive courses to the same group, etc. In such cases, hard and soft constraints should be satisfied and the timetabling will be a global hard integer constraint problem which can be expressed as follows:

$$\sum_{i=1}^{|\pi|}\sum_{j=1}^{|\chi|}\sum_{k=1}^{|\mu|}\sum_{l=1}^{|\varepsilon|}\sum_{s=1}^{|\rho|}\sum_{m=1}^{|\eta|}\sum_{n=1}^{|\delta|} X_{C\times T}(i,j,k,l,s,m,n)=1 \quad (1)$$

In (1), $X_{C\times T}(i,j,k,l,s,m,n)=1$ represents the number of students (group) of the i^{th} and the section j taking course k with the teacher l in the classroom s during the time slot n of the day m. The product *CxT* allows distinguishing between ordinary courses taking one hour timeslot denoted by "*C*" and those taking two hours timeslots like sport and industrial workshop activities denoted by "*T*" and its value is clearly indicated by *n/n+1* in the equation (9). In order to explain our methodology, we will give an informal presentation of the timetabling problem as follows:

Assume that we have three levels of students (first, second and third year) and seven groups of students. We define then the indexes as follows:

* $i=1\rightarrow$ first year, $i=2\rightarrow$ second year, $i=3\rightarrow$ third year,
* $j=1,2,3,4,5,6,7\rightarrow$ Represents the sections (classes) namely, $A_1,A_2,A_3,A_4,A_5,A_6,A_7$.
* $k=1\rightarrow Maths, k=2\rightarrow Arab, k=3\rightarrow Natural-Science,$ $k=4\rightarrow Physics, k=5\rightarrow Geography,$
* $k=6\rightarrow French, k=7\rightarrow English,$ $k=8\rightarrow Sports, k=9\rightarrow Industrial-design$

We write: $X_C(i,j)=1$ if the section (i,j) exists and $X_C(i,j)=0$ otherwise. We note that $\sum_{j=1}^{7} X_C(i,j)=\overline{C}(i)$ is the total number of the classes in the year i, for instance $\overline{C}(1)=6,\overline{C}(2)=6,\overline{C}(3)=7$ and the number of all the classes is $\sum_{i=1}^{3}\sum_{j=1}^{7} X_C(i,j)=\sum_{i=1}^{3}\overline{C}(i)=19$.

Let $X_M(i,j,k)=1$ if the group (i,j) and the course k exists with a global duration equals to, $a(i,j,k)$. We write $\sum_{k=1}^{9} X_M(i,j,k)=\overline{M}(i,j)=9$ the number of courses for the group $X_C(i,j)$ ($\forall(i,j) i=1,2,\cdots,$ $j=1,2,\cdots$. For instance, $\overline{M}(1,7)=0,$ $\overline{M}(3,j)=8, j=1,\cdots,6$. $\sum_i\sum_j\sum_{k=1}^{9} X_M(i,j,k)=\sum_i\sum_j \overline{M}(i,j)=12\times 9+7\times 8=164$ combinations ($classes\times courses$).

Definition: Let (i,j) be a class, k a lecture, l a teacher, s a classroom, m a given day of the week and n a timeslot in the day m, we write then:

$$X_C(i,j,k,l,s,m,n)=1 \quad (2)$$

and we say that relation (2) is valid if the teacher l of the lecture k is assigned to the class (i,j) in the day m for the timeslot n. Otherwise we write the following relation:

$$X_C(i,j,k,l,s,m,n)=0 \quad (3)$$

Note that the working days of the week are set to six labelled by the integer numbers from 1 to 6 where the first day of the week is Monday (label 1) and the second day is Tuesday (label 2), on so. We consider that there are 10 one hour timeslots in a day denoted by 1 to 10 (beginning from 08h00 past midnight to 06h00 past midday), except that the last day of the week (Friday), which has 4 timeslots only (beginning from 08h00 AM to 12h00 AM).

The logical constraints are depicted as follow:

* Every teacher is assigned to at most only one course with the class (i,j) in the day m at the period n.

$$\sum X_C(i,j,k,l,s,m,n)+\overline{d}_C^{\,1}(i,j,k,l,s,m,n)=1 \quad (4)$$

where $\overline{d}_C$ is a Boolean deviation variable belonging to the set $(0,1)^k$.

* Every teacher is assigned to at most a lab course with the class (i,j), in the day m taking two consecutive timeslots *n/n +1*.

$$\sum X_T(i,j,k,l,s,m,n/n+1)+\overset{-1}{d_T}(i,j,k,l,s,m,n)=1 \quad (5)$$

* Let $a_t(i,j,k,l,s,m,n)$ be the duration of the assignment $X_C(i,j,k,l,s,m,n)$. To ensure that for every class (i,j) with a course k, the duration of an ordinary course or a lab course is respected, we have to verify the validity of the following equations:

$$\sum_l\sum_m\sum_n X_C(i,j,k,l,s,m,n)\times a_c(i,j,k,l,s,m,n)+\overset{-2}{d_C}(i,j,k)-\overset{+2}{d_C}(i,j,k)=a_c(i,j,k) \quad (6)$$

and

$$\sum_l\sum_m\sum_n X_T(i,j,k,l,s,m,n)\times a_T(i,j,k,l,s,m,n)+\overset{-2}{d_T}(i,j,k)-\overset{+2}{d_T}(i,j,k)=a_T(i,j,k) \quad (7)$$

and

$$a_C(i,j,k)+a_T(i,j,k)+\overset{-3}{d}(i,j,k)-\overset{+3}{d}(i,j,k)=a(i,j,k) \quad (8)$$

* Every teacher is assigned to at most a number of timeslot equals to the value $\overline{E}(l)$ for each week:

$$\sum_{i,j,l,s,m,n} X_C(i,j,k,l,s,m,n)+\sum_{i,j,l,s,m,n} X_T(i,j,k,l,s,m,n/n+1)+\overset{-4}{d}(l)-\overset{+4}{d}(l)=\overline{E}(l)$$

$$l=1,2,\cdots,38 \quad (9)$$

* We associate with each class (i,j) a classroom $S(i,j)$ (k=3,4). Thus, the maximal timeslots number available for each classroom is the following:

$\bar{S}(i,j,m)=7$ for m=1,2,4,5 $\forall(i,j)$ and

$\bar{S}(i,j,m)=4$ for $m=3,6$ $\forall(i,j)$

$$\sum_{k,l,n} X_C(i,j,k,l,s,m,n)+\sum_{i,j,k} X_T(i,j,k,l,s,m,n)\times 2+\overset{-5}{d}(i,j,m)=\bar{S}(i,j,m) \quad (10)$$

$$\sum_{k,l,m,n} X_C(i,j,k,l,s,m,n)+\sum_{k,l,m,n} X_T(i,j,k,l,s,m,n/n+1)\times 2+\overset{-6}{d}(i,j,m)=\sum_{m=1}^{6}\bar{S}(i,j,m) \quad (11)$$

* For the workshop and the laboratory classrooms (k=3,4), we define the following relations:

$XT_{G1}(i,j,k,l,m,n/n+1)$ and $XT_{G2}(i,j,k,l,m,n/n+1)$ (12)

$XT_{G2}(i,j,3,l,m,n/n+1)=0$ and
$XT_{G2}(i,j,4,l,m,n/n+1)=1$ *if*
$XT_{G2}(i,j,3,l,m,n/n+1)=1$
$XT_{G1}(i,j,3,l,m,n/n+1)=0$ and
$XT_{G1}(i,j,4,l,m,n/n+1)=1$ *if*
$XT_{G2}(i,j,3,l,m,n/n+1)=1$
$XT_{G2}(i,j,4,l,m,n/n+1)=0$ and
$XT_{G2}(i,j,3,l,m,n/n+1)=1$ *if*
$XT_{G1}(i,j,4,l,m,n/n+1)=1$
$XT_{G1}(i,j,4,l,m,n/n+1)=0$ and
$XT_{G1}(i,j,3,l,m,n/n+1)=1$ *if*
$XT_{G2}(i,j,4,l,m,n/n+1)=1$ (13)

* For the laboratories (with k = 3), we have the following:

$$\sum_j\sum_l\sum_n XT_{G1}(i,j,3,l,m,n/n+1)\times 2+\sum_j\sum_l\sum_n XT_{G2}(i,j,3,l,m,n/n+1)\times 2$$
$$+\overset{-7}{d_T}(i,3,m)-\overset{+7}{d_T}(i,3,m)=\overline{L}_i(m) \text{ for } (i=1,2,3) \quad (14)$$

* For the workshops, we have the following:

$$\sum_j\sum_l\sum_n XT_{G1}(i,j,4,l,m,n/n+1)\times 2+\sum_j\sum_l\sum_n XT_{G2}(i,j,4,l,m,n/n+1)\times 2$$
$$+\overset{-7}{d_T}(i,4,m)-\overset{+7}{d_T}(i,4,m)=\overline{A}_i(m) \quad (15)$$

* Generalisation (for the week):

$$\sum_j\sum_l\sum_m\sum_n XT_{G1}(i,j,3,l,m,n/n+1)\times 2+\sum_j\sum_l\sum_m\sum_n XT_{G2}(i,j,3,l,m,n/n+1)\times 2$$
$$+\overset{-8}{d_T}(i,3)-\overset{+8}{d_T}(i,3)=\sum_{m=1}^{6}\overline{L}_i(m) \quad (16)$$

$$\sum_j\sum_l\sum_m\sum_n XT_{G1}(i,j,4,l,m,n/n+1)\times 2+\sum_j\sum_l\sum_m\sum_n XT_{G2}(i,j,4,l,m,n/n+1)\times 2$$

$$+\overset{-8}{d_T}(i,4)-\overset{+8}{d_T}(i,4)=\sum_{m=1}^{6}\overline{A}_i(m) \text{ for } i=1,2,3 \quad (17)$$

* We want to assign at least a lab course of two timeslots duration to a given teacher, in the morning of each week. We write the following:

$$\sum_{m=1}^{6}\sum_{n=1}^{3} X_T(i,j,1,l,m,s,n/n+1)+\overset{-9}{d_T}(i,j,1,l)-\overset{+9}{d_T}(i,j,1,l)=1 \quad \forall(i,j,l) \quad (18)$$

$$\sum_{m=1}^{6}\sum_{n=1}^{3} X_T(i,j,1,l,s,m,n/n+1)\times a(i,j,1,l)+2\times\left[\overset{-10}{d_T}(i,j,1,l)-\overset{+10}{d_T}(i,j,1,l)\right]=2 \quad \forall(i,j,l) \quad (19)$$

* For the mathematical courses, they should be taught only in the morning periods, thus we write the following:

$$\sum_{m=1}^{6}\sum_{n=1}^{4} X_C(i,j,1,l,s,m,n)+\overset{-11}{d_C}(i,j,1,l)-\overset{+11}{d_C}(i,j,1,l)=a_C(i,j,1,l)\quad \forall(i,j,l) \tag{20}$$

* If we sum that the last equation using *l*, we get the following:

$$\sum_{l=1}^{7}\sum_{m=1}^{6}\sum_{n=1}^{4} X_C(i,j,1,l,s,m,n)+\overset{-12}{d_C}(i,j,1)-\overset{+12}{d_C}(i,j,1)=a_C(i,j,1)\quad \forall(i,j) \tag{21}$$

and using (i, j) we obtain:

$$\sum_{i}^{3}\sum_{j=1}^{7}\sum_{l=1}^{7}\sum_{m=1}^{6}\sum_{n=1}^{4} X_C(i,j,1,l,s,m,n)+\overset{-13}{d_C}-\overset{+13}{d_C}=a_C\quad \forall(i,j) \tag{22}$$

* The timeslots of the sports activity units are consecutives, it suffices to proceed exactly like with the lab courses:

$$\sum_{m=1}^{6}\sum_{n=1}^{6} X_T(i,j,k=8,l=36,s,m,n/n+1)\times 2+2\times\overset{-14}{d_T}=2\quad \forall(i,j) \tag{23}$$

$$\sum_{m=1}^{6}\sum_{n=1}^{6} X_T(i,j,k=8,l=37,s,m,n/n+1)\times 2+2\times\overset{-15}{d_T}=2\quad \forall(i,j) \tag{24}$$

* We must minimize the number of free timeslots not occupied by a teacher during a day of the week. Let *l* be a teacher and $N_C(k,l,m)$ + $N_T(k,l,m)$ be the number of timeslots occupation during a day, we write:

$$\sum_{i}\sum_{j}\sum_{n} X_C(i,j,k,l,s,m,n)=N_C(k,l,m) \text{ and}$$
$$\sum_{i}\sum_{j}\sum_{n} X_T(i,j,k,l,s,m,n/n+1)=N_T(k,l,m) \tag{25}$$

We know that

$$\sum_{k} X_C(i,j,k,l,m,n)+\overset{-1}{d_C}(i,j,l,m,n)=1 \quad \textit{and} \quad \sum_{k} X_T(i,j,k,l,s,m,n/n+1)+\overset{-1}{d_T}(i,j,l,s,m,n/n+1)=1 \tag{26}$$

and because, each teacher is not assigned to more than one course with a class (i, j), we write the following relation:

$$X_C(i,j,k,l,m,n)+\overset{-16}{d_C}(i,j,l,m,n)=1 \quad \textit{and} \quad X_T(i,j,k,l,m,n/n+1)+\overset{-16}{d_T}(i,j,l,m,n/n+1)=1 \tag{27}$$

* We define $\alpha_{k,l,m} \le \gamma$ for a day *m* and $n = 1,2,6$:

$$\alpha_C(i,j,k,l,m) \le \gamma_c(n)\left[1-\overset{-1}{d_c}(i,j,k,l,m,n)\right]$$

where $\gamma_c(n)=1,\cdots,7 \; for \quad n=7,\cdots,1$ (28)

and

$$\beta_C(i,j,k,l,m) \ge \delta_c(n)\left[1-\overset{-17}{d_c}(i,j,k,l,m,n)\right] \tag{29}$$

$\delta_c(n)$ is defined exactly like $\gamma_c(n)$. However, the minimization of the relation (30) is needed,

$$\sum\sum\alpha_C(i,j,k,l,m)+\sum\sum\beta_C(i,j,k,l,m)-\mathrm{M}(n)=\Gamma_c(k,l,m) \tag{30}$$

Similarly, to minimize the lost time for the lab courses, we define the constraints (31) and (32),

$$\alpha_T(i,j,k,l,m) \le \gamma_T(n)\left[1-\overset{-17}{d_T}(i,j,k,l,m,n/n+1)\right] \tag{31}$$

Where $\gamma_c(n)=1,\cdots,5 \; for \quad n=5,\cdots,1.$

$$\beta_T(i,j,k,l,m) \ge \delta_T(n)\left[1-\overset{-18}{d_T}(i,j,k,l,m,n/n+1)\right] \tag{32}$$

Like in the relation (30), we construct the following equation:

$$\sum_{i}\sum_{j}\alpha_T(i,j,k,l,m)+\sum_{i}\sum_{j}\beta_T(i,j,k,l,m)-\mathrm{M}(n)=\mathrm{T}_{TP}(k,l,m)$$
$$\forall(k;l;m) \tag{33}$$

By combining the relations (30) and (33), we need to minimize the following equation:

$$\sum_{i}\sum_{j}\alpha_C(i,j,k,l,m)+\sum_{i}\sum_{j}\alpha_T(i,j,k,l,m)+\sum_{i}\sum_{j}\beta_C(i,j,k,l,m)+\sum_{i}\sum_{j}\beta_T(i,j,k,l,m)-2\times\mathrm{M}(n)=\mathrm{T}(k,6,m) \tag{34}$$

Assume that $S_m(i, j)$ is the available time of a classroom S(i, j) during a day *m*, so:

$$\sum_{k}\sum_{l}\sum_{n}^{M(n)} X_C(i,j,k,l,s,m,n)+\sum_{k}\sum_{l}\sum_{n}^{M(n)} X_T(i,j,k,l,s,m,n/n+1)\times 2+\overset{-19}{d_m}(i,j)=S_m(i,j,1)\quad \forall(i,j) \tag{35}$$

More generally, if we sum on *m*, we minimize the dead time of a classroom using the following constraint:

$$\sum_{k}\sum_{l}\sum_{m=1}^{6}\sum_{n=1}^{M(n)} X_C(i,j,k,l,s,m,n)+\sum_{k}\sum_{l}\sum_{m=1}^{6}\sum_{n}^{M(n)} X_T(i,j,k,l,s,m,n/n+1)\times 2+\overset{-20}{d_m}(i,j)=\sum_{m=1}^{6} S_m(i,j,1)\quad \forall(i,j) \tag{36}$$

* We want, that each teacher will be free at least one day in the week; Let *l* be such a teacher, we interpret such situation by the following constraint:

$$\sum_{m=1}^{\lambda}\left[\sum_{i,j,k,n} X_C(i,j,k,l,s,m,n)+\sum_{i,j,k,n} X_T(i,j,k,l,s,m,n/n+1)\times 2\right]=L_\lambda(l) \tag{37}$$

We define then:

$$\sum_{i,l,k,n} X_C(i,j,k,l,s,m,n) + \sum_{i,l,k,n} X_T(i,j,k,l,s,m,n/n+1) \times 2 = N(l,m) \quad (38)$$

and we write:

$$N(l,m) - \overset{-16}{d}(l,m) = 0 \quad m = 1,2,\cdots,6 \quad (39)$$

It is expected to minimise the occupation (number of timeslots) by each teacher:

$$\text{Minimise } \sum_{m=1}^{6} \overset{-21}{d}(l,m)$$

$$\text{Subject to } \sum_{m=1}^{6} N(l,m) = \overline{N}(l) \quad (40)$$

Finally, the timetabling solution is given after solving the following linear system:
Minimization of the following function:

$$\left(\sum_{i,j,k}\overset{-2}{d_C}(i,j,k)+\sum_{i,j,k}\overset{+2}{d_C}(i,j,k)\right)+\left(\sum\overset{-2}{d_T}(l)+\sum\overset{+2}{d_T}(l)\right)+\left(\sum_{i,j,k}\overset{-3}{d}(i,j,k)+\sum_{i,j,k}\overset{+3}{d}(i,j,k)\right)+\left(\sum_{l}\overset{-4}{d}(l)+\sum_{l}\overset{+4}{d}(l)\right)$$
$$+\left(\sum_{i,j,m}\overset{-5}{d}(i,j,m)+\sum_{i,j,m}\overset{+5}{d}(i,j,m)\right)+\left(\sum_{i,j,m}\overset{-6}{d}(i,j,m)+\sum_{i,j,m}\overset{+6}{d}(i,j,m)\right)+\left(\sum_{i,m}\overset{-7}{d_C}(i,3,m)+\sum_{i,m}\overset{+7}{d_C}(i,3,m)\right)$$
$$+\left(\sum_{i,m}\overset{-7}{d_T}(i,4,m)+\sum_{i,m}\overset{+7}{d_T}(i,4,m)\right)+\left(\sum_{i}\overset{-8}{d_T}(i,3)+\sum_{i}\overset{+8}{d_T}(i,3)\right)+\left(\sum_{i}\overset{-8}{d}(i,4)+\sum_{i}\overset{+8}{d}(i,4)\right)+$$
$$\left(\sum_{i,j,l}\overset{-9}{d}(i,j,1,l)+\sum_{i,j,l}\overset{+9}{d}(i,j,1,l)\right)+\left(\sum_{i,j,l}\overset{-10}{d_C}(i,j,1,l)+\sum_{i,j,l}\overset{+10}{d_C}(i,j,1,l)\right)+\left(\sum_{i,j}\overset{-11}{d_C}(i,j,1,l)+\sum_{i,j}\overset{+11}{d_C}(i,j,1,l)\right)$$
$$+\left(\sum\overset{-12}{d_C}(i,j,1)+\sum\overset{+12}{d_C}(i,j,1)\right)+\left(\sum\overset{-13}{d_C}+\sum\overset{+13}{d_C}\right)+\left(\sum\overset{-14}{d_T}+\sum\overset{+14}{d_T}\right)+\left(\sum\overset{-15}{d_T}+\sum\sum\overset{+16}{d}(l,m)\right)+$$
$$\left(\sum\overset{-1}{d_C}(i,j,l,m,n)+\sum\overset{-1}{d_T}(i,j,l,m,n/n+1)\right)+\left(\sum_{k,l,m}\left[\sum_{i}\sum_{j}\alpha_C(i,j,k,l,m)+\sum\sum\overline{\alpha}_T(i,,k,l,m\right]\right)$$
$$+\sum_{i}\sum_{j}\overset{+}{\beta}_C(i,j,k,l,m)+\sum_{i}\sum_{j}\beta_T(i,,k,l,m)$$

under the constraints from (4) to (28) and the constraint (35).

ALGORITHMES

The sequential construction algorithm introduced by Asmuni *et al.*[16], is applied to build a timetable solution once a preliminary scheduling was ordered using a random or a fuzzy expert system. The algorithm is depicted in an exhaustive form as given below.

The algorithm requires some steps to assign the courses to the available timeslots. First, the lectures are ordered by some criteria selected by the decision maker according to the preferences objectives, for instance, the first two slots of the first day of the week are assigned to Maths courses. The courses are selected in a descending order using a weighted function where the scientific courses (e.g. Maths, Physics, Programming, etc.) get the highest values. Then the courses are assigned to the last available timeslot in the list with a minimum penalty cost. If no free timeslot is available,

Algorithm 1:
// Data preparation could be done randomly or using a fuzzy engine
Begin
  Sort the unscheduled lectures using the selected preferences ordering criteria;
  Choose a day of the week;
 // the choice could be done according to a random criteria
Do While (the expected days have not been selected)
   Begin
    Insert a lecture into the first timeslot of the target day;
   Do While (there exist a free timeslot)
    Perform the process for scheduling the unscheduled lectures;
    Sort the unscheduled lectures using the selected preference ordering criteria;
    Insert a lecture into the next timeslot of the target day;
   End dowhile
Choose the next day;
End
End dowhile
End.

Fig. 1: Pseudo code for general data scheduling

Algorithm 2:
Select the filled timetable
If all the conflicts are removed, thus the timetabling is valid;
Stop the process and Print the timetable.
Otherwise
Create the system of constraints and the multiple deviations functions;
Call the linear programming SOLVER;
Iterate the process until a solution is found;
If it is a good solution, then stop the process and print the timetable;
Otherwise
Call the first Algorithm (Fig 1).
End
End.

Fig. 2: Pseudo code for generating the timetable solution

the course is skipped and the process continues with the next courses. The skipped courses are then revisited and a process for re-scheduling them will be performed as shown in algorithm 1 (Fig. 1). Once, the steps are executed and a valid timetable is not yet generated, the following algorithm will be applied.

CONCLUSION

In this research, we have shown how the combinatorial timetabling problem can be modelled and solved using goal programming techniques. As far as we know, no published work has used goal programming in modelling and solving the timetabling problem. Our experiments carried out using a college of 1000 students, 38 teachers, 3 different levels, 19 classical classrooms, 3 laboratory classrooms, 3 workshop rooms and a sport room have shown that goal programming is a promising technique to generate optimal solutions whenever the lectures are ordered by several criteria before the search process starts. Future research work may include the followings:

* Investigating other combinations constraints
* Investigating different sets of fuzzy rules and fuzzy membership functions
* Exploring other optimisation algorithms

ACKNOWLEDGEMTN

We would like to thank Professor GV Sarma for his helps given during the period of 1986-1992 when he was the research advisor at the department of Computer Sciences of the Annaba University, Algeria. We think that he has influenced a great part of the research contained in this paper

REFERENCES

1. Abdennader, S. and M. Marte, 1999. University timetabling using constraint handling rules. J. Applied Artif. Intel., Special Issue on Constraint Handling.
2. Merlot, L., N. Boland, H. Hughes and P. Stuckey, 2002. A hybrid algorithm for the examination timetabling problem. In: 4th Intl. Conf. Practice and Theory of Automated Timetabling, 21-23 Aug. 2002. PATAT 2002.
3. Easton, K., G. Nemhauser and M. Trick, 2002. Solving the travelling tournament problem: A combined integer programming and constraint programming. In the Proc. 4th Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '02).
4. Erben, W. and J. Keppler, 1995. A genetic algorithm solving a weekly course-timetabling problem. In the Proc. of the First Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '95).
5. Michalewicz, Z. and D.B. Fogel, 2000. How to Solve It- Modern Heuristics. Springer.
6. Burke, E. and J. Newall, 1999. A multi-stage evolutionary algorithm for the timetable problem. J. IEEE Trans. Evolutionary Computation, 3: 63-74.
7. Burke, E.K., Y. Bykov and S. Petrovic, 2000. Multicriteria approach to timetabling problems. In the Proc. Third Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '00).
8. Thomson, J. and K. Dowsland, 1995. General cooling schedules for simulated annealing-based timetabling systems. In. Proc. First Intl. Conf. Practice and Theory of Automated Timetabling, pp: 345-363.
9. Schaerf, A., 1996. A survey of automated timetabling. Technical Report CS-R9567, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands.
10. Schaerf, A., 1996. Tabu search techniques for large high-school timetabling problems. In Proc. Fourteenth National Conf. on Artificial Intelligence, pp: 363-368, Portland, Oregon, USA.
11. Elkhyari, A., C. Guéret and N. Jussien, 2003. Solving dynamic timetabling problems as dynamic resource constrained project scheduling problems using new constraint programming tool. Practices and Theory of Automated Timetabling IV, Lecture Notes in Computer Science, 2740, Springer-Verlag, pp: 39-59.
12. Elkhyari, A., C. Guéret and N. Jussien, 2004. Stable solutions for dynamic project scheduling problems. In PMS'04 Intl. Workshop on Project Management and Scheduling, Nancy, France, pp: 380-384.
13. Muller, T., 2002. Interactive heuristic search algorithm. In Proc. Doctorate Programme of CP-2002 Conf., Ithaca.
14. Schaerf, A., 1999. A survey of automated timetabling. Artif. Intell. Rev., 13: 87-127.
15. Asmuni, H., E.K. Burke and J.M. Garibaldi, 2004. Fuzzy multiple ordering criteria for examination timetabling. In the Proc. 5th Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '04).
16. Chand, A., 2004. A constraint based genetic model for representing complete university timetabling data. In the Proc. 5th Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '04).
17. Marte, M., 2000. Towards constraint-based grammar school timetabling. In the Proc. Third Intl. Conf. Practice and Theory of Automated Timetabling (PATAT '00).
18. Charnes, A., W.W. Cooper and R. Ferguson, 1955. Optimal estimation of executive compensation by linear programming. Manag. Sci., 1: 138-151.
19. Charnes, A. and W. Cooper, 1961. Management Models and Industrial Applications of Linear Programming. Vols. I and II, John Wiley and Sons, New York.
20. Chang Ni-Bin and Y.L. Wei, 2000. Sitting recycling drop-off stations in urban area by genetic algorithm-based fuzzy multiobjective nonlinear integer programming modelling. Fuzzy Sets and Systems, 114: 133-149.
21. Can, E. and M. Houck, 1983. Real-time reservoir operations by goal programming. J. Water Resources Plan. Manag., ASCE, 110: 297-309.

22. Loganathan, G.V. and D. Bhattacharya, 1990. Goal-programming techniques for optimal reservoir operations. ASCE J. Water Resources Plan. Manag., 116: 820- 838.
23. Zagona, E., T. Fulp, H.M. Goranflo and R. Shane, 1998. Riverware: A general river and reservoir modelling environment. In Proc. First Federal Interagency Hydrologic Modelling Conf., Las Vegas, NV, pp: 5-113.
24. Plastria, F., 1995. Continuous location problems. In Facility Location. A Survey of Applications and Methods. Springer-Verlag, New York, pp: 225-262.
25. Tamiz, M., D. Jones and C. Romero, 1998. Goal programming for decision making: An overview of the current state-of-the-art. Eur. J. Oper. Res., 11: 569-581.
26. Dauer, J. and Y.H. Liu, 1997. Multi-Criteria and Goal Programming. In Advances in Sensitivity Analysis and Parametric Programming (Eds. T. Gal and H.J. Greenberg.), Kluwer, Boston.
27. Schniederjans, M.J., 1995. Goal Programming, Methodology and Applications. Kluwer, Boston.