A MATHEMATICAL FORMULATION FOR A SECONDARY SCHOOL TIMETABLING PROBLEM (*)

Margarida Vaz Pato (**)
Marco Paulo Carrasco (***)

1 — Introduction

At teaching institutions of all levels, from kindergardens to universities, including institutions providing isolated courses, the task of creating timetables is of major importance. Timetables are needed to schedule regular courses, exams and special events, such as short courses, conferences [Eglise and Rand (1987)] and sports events. However, each operation differs considerably — even when scheduling the same kind of event — among schools of the same type. A survey of classic timetabling models may be found in de Werra (1984), and recent references on the Internet [Carter (1995)].

Generally speaking, in schools two main kinds of timetables arise: timetables for examinations and timetables for classes provided throughout the year.

The first type — *exam timetabling* — is related to the assigning of exams to time slots, for each group of students with the same curriculum, in such a way that the exam period is respected and tests cover the entire allocated exam period [Carter, Laporte and Chinneck (1994), Laporte and Desroches (1984) and Johnson (1990)].

The second type — *class timetabling* — concerns the production of timetables for students and teachers in keeping with the curricula, available rooms and compliance with certain rules determining timetabling quality.

From the computational perspective, most of the problems relating to the two types are complex [Even and Shamir (1976), Cooper and Kingston (1995)]. However, class timetabling is far more intricate as it deals with many more entities and is subject to countless rules.

The problem we shall present in this paper concerns class timetabling at a secondary school in Portugal, the so-called secondary school class timetabling problem — STP for short.

In section 2 a brief review is made of similar class timetabling problems, together with a reference to the respective formulations and methods used by authors to solve them.

Section 3 is devoted to a description of the STP, whose formulation is given in Section 4. Section 5 provides an illustrative example taken from an unrealistic, very small school, and Section 6 completes our report with several relevant comments.

(*) This research has been partly supported by the Centro de Investigação Operacional (FC — Universidade de Lisboa) within the PRAXIS XXI (JNICT) Project no. 2/2.1/MAT/139/94.

(**) Instituto Superior de Economia e Gestão, Universidade Técnica de Lisboa.

(***) Escola Superior de Gestão, Hotelaria e Turismo, Universidade do Algarve.

2 — A review of European secondary school class timetabling

The class timetabling problem depends on the specific educational system one is dealing with. As a rule, European countries possess a knowledge system that differs considerably from that of North America, including Canada [Ferland and Roy (1985), Hertz (1991)], though some European universities do share the main timetabling characteristics of the overseas system [Downsland (1990)].

In the American system the curricula are less rigid than in Europe; students are allowed to enroll in optional subjects and schedules ensure that clashes are minimized. However, in Europe optional subjects are the exception in the curricula, and timetabling clashes are not contemplated at all. Moreover, in European schools, timetables are generally performed on a weekly basis, whereas in the American system timetables operate on a daily basis. This, to a large extent, explains why overseas models and software for timetabling cannot be used in our schools.

Here we explain some European models of interest which are in some way related to our case study — the STP.

Most literature published to date concerning timetabling addresses a wide range of solution methodologies to tackle the problems: network flows, matching, project scheduling with resource constraints, optimal coloring, decomposition and branch-and-bound algorithms, lagrangean relaxation techniques and heuristics — constructive, local search or meta-search.

Let us begin by briefly explaining a timetabling problem found in the survey by de Werra (1984).

The basic model calls for a set of classes — groups of students with the same curriculum —, a set of teachers and a matrix whose elements represent the number of lessons involving each class and each teacher. The lessons are of equal length (time slot) and a period of p time slots is submitted for scheduling. The solution to the timetabling problem will assign each lesson to one time slot to ensure that no teacher and no class is involved in more than one lesson at a time. The author formulates this problem as a system of linear constraints, whose binary variables $xijk(s)$ become 1 if class i and teacher j are assigned to the same lecture at time slot k, and 0 if otherwise.

This author also formulates the above problem as one of edge coloring in a bipartite multigraph, where the nodes stand for classes and teachers, and parallel edges stand for the lessons that involve each teacher/class pair.

Both formulations are extended to cover the weekly timetables. Theoretical results are given for the problems and heuristic methods are presented, on the basis of network flows and matchings.

Colorni, Dorigo and Maniezzo (1990) tackle a class timetabling problem whose data consists of:

A set of teachers;
A set of classes;
A set of weekly teaching time slots for each class;
The curriculum for each class based on the attendance of the teachers working for that class;
Other conditions, such as personal preferences of the teaching staff, pedagogical and organizational rules.

Timetables must meet the predefined number of lessons for every teacher, as well as for each class and respective constraints, i. e.:

- There should be no more than one teacher in a class in any time slot;
- No teacher may teach more than one class in one time slot;
- There must be one teacher for each class scheduled for a specific time slot.

When building timetables, one endeavours to cater for teachers' personal preferences, besides organizational and the pedagogical rules, such as the spreading of lessons over the day periods.

The above authors use a genetic search representing each solution of the timetabling problem by a chromosome in matrix form. Each row of the matrix stands for one teacher, while each column represents a time slot. The values of the matrix elements are classes.

Genetic operators of selection, mutation, crossover and filtering were specifically developed for this case. Some constraints are imposed by penalizing the fitness of the chromosomes, while others are imposed through the genetic operators.

It is known that this methodology was successfully applied at a large high school in Milan.

We shall now briefly review a paper due to Abramson (1991), who uses a simulated annealing-based heuristic. We found another reference, Abramson and Abela (1991), which contains a genetic algorithm for a class timetabling problem, but have since been unable to obtain this document.

In this problem the authors assume that classes, teachers and rooms have already been scheduled in accordance with the teachers' preferences and available rooms. The problem of creating timetables then becomes one of assigning the triplets (class/room/teacher) to a set of time slots, while respecting the following hierarchically defined conditions that determine the quality of the timetable:

- Avoid the simultaneous occurrence of the triplets (class/room/teacher) that concern the same resources;
- When optional subjects are involved, avoid the simultaneous occurrence of classes with common students;
- Choose rooms suited for each subject;
- Allow multi-slot-time classes;
- Bound the number of lessons, both for students and teachers, within the specific time period — e. g. a week.

To enable such guidelines to be incorporated in the model, an objective function, composed of different parcels, is created. This function penalizes violation of the above conditions and takes value zero when all conditions are verified.

The algorithm begins with a random solution and is followed by an annealing procedure defined in a specific neighbourhood. In each iteration, a time slot and a triplet (class/room/teacher) are randomly selected. Another time slot is then randomly chosen to perform a move (towards a new solution) and the change in respective objective value provided by the move is calculated. Adoption or otherwise of the move depends on the probability guiding the annealing scheme.

An empirical study is then presented with tests of different annealing parameters in a set of problem instances, including randomly generated instances and a real data instance from a secondary school in Australia.

The paper by Gislén, Peterson and Soderberg (1989) presents an application of neural networks to a class timetabling problem and later, *in* 1992, a further paper by the same authors perfects the methodology.

The authors analyse the four sets of entities usually involved in timetabling situations (teachers, classes, rooms and time slots) and work with two types of pairs: teacher/class and room/slot. The problem thus becomes one of assigning pairs of one type to pairs of the other type, while simultaneously respecting problem constraints.

The authors apply a neural network by using Potts' neurons and mean field theory with an annealing scheme to enforce the convergence to better quality solutions. Each neuron represents an assignment of one teacher/class pair to a room/slot pair. The neurons begin by being given a real value, whereas when the neural procedure ends, they converge to become binary values, 1 or 0. This work explores the idea of separating the conditions into two obligatory hierarchical grades (hard and soft constraints) embedding all constraints in the energy function of the neural network by penalizing them with different parameters.

Two other recent papers refer to the devising of timetables involving sets of teachers, classes, subjects and rooms.

One is due to Costa (1994) and carefully reports on a very complete problem of timetabling in high/secondary schools in Switzerland. The problem is formulated as a nonlinear binary problem embedding some constraints (the soft ones) in the objective function and others (the hard) as simple constraints. The solution procedure is a specially designed tabu search which was successfully tested in two real cases from the model's field of application.

The other paper is from Alvarez-Valdes, Martin and Tamarit (1995). Using the above elements, the authors consider the problem as a resource-constrained project scheduling one. The lessons constitute the activities to be scheduled, taking into account the limited resources given by the classes, teachers and rooms, whereas the precedence constraints between the lessons of the same subject generate the scheduling constraints. A tabu search heuristic is also applied to improve the above feasible solution. The overall method is tested on a set of 14 Spanish high schools and in high dimensioned instances obtained from random generation.

Let us now refer to a timetabling problem successfully applied to a secondary school in the United Kingdom and due to Write (1996). This case is very specific to the actual school concerned. The timetabling covers a fortnight and considers the subjects clustered into blocks. As usual in this field, some of the constraints are taken as hard constraints, that is they are regarded as constraints peculiar to the mathematical programming model, whereas others are taken as soft. In other words they are embedded as penalizing terms in the objective function. The process of creating timetables is performed in stages, some of which are computational, while others are manual.

3 — The secondary school class timetabling problem

Let us now explain our timetabling problem which, as mentioned above, was inspired on a specific secondary school. It does, however, resemble most secondary class timetabling problems to be found in our country.

The overall timetabling process in this kind of school may be broken down into four stages, as shown in figure 1.

FIGURE 1

The overall timetabling process

Stage 1 — student enrollment Stage 2 — class building Stage 3 — assignment of classes to teachers Stage 4 — creation of timetables

Firstly students enroll and choose the optional subjects they prefer, assuming the curriculum contains options.

Classes are then constructed by taking into account both optional subjects and advisable sizes. To be more precise, a class is built with students sharing the same basic curriculum. There may, however, be some optional subjects that require a class to be split and/or merged into other classes, thus forming groups of students attending a specific subject.

In the third stage, classes are assigned to teachers while respecting teacher specialization and respective weekly work-load. Teachers are assigned to classes via subjects and, as a result, teacher/class(es) pairs are established. These we shall refer to as *disciplines*. For example, if Teacher 1 teaches Mathematics to Class A and Class B separately, we define the corresponding teacher/class pairs as disciplines Math 1A and Math 1B. Also, if Teacher 2 teaches the optional subject German to a group of students from Class A and Class B, there is also a teacher/class pair defining discipline Germ 2AB. It should be added that a subject for a specific group of students cannot be taught by different teachers.

The fourth stage involves creating the timetables — which we shall be addressing throughout the rest of this paper and is referred to as the secondary school class timetabling problem.

Solving this problem entails building weekly timetables for all *classes, teachers* and *rooms.* Each week is made up of days — let us say 5 in most cases — and the number of *time slots* per day is known. It is assumed that each lesson occupies only one time slot.

The number of constraints to be satisfied is considerable and are, in themselves, complex. In handling these difficulties, two hierarchical levels of constraints were considered: level 1 obligatory or compatibility constraints and level 2, improvement constraints.

Level 1 constraints must be satisfied in any feasible solution and include the following restrictions:

Each room must be occupied in each time slot by no more than one discipline;

Each teacher must teach no more than one discipline in each time slot;

Each student must attend no more than one discipline in each time slot.

The last condition applies to the subsets of students with different options. For instance, one class may have students taking French and others taking English as a foreign language. In such cases two different disciplines are assigned to that class to include the foreign languages and the constraint is enforced for the two subgroups of students.

Other level 1 constraints refer to specific aspects of the problem, such as the weekly number of time slots per subject, as stipulated by the curriculum of the students, room capacity and room suitability for each subject.

Pedagogical rules are also considered, such as the enforcement of a lunch break, both for teachers and students. A schedule that respects a maximum number of daily lessons per discipline is also strongly advisable.

The group of so called *level 2 constraints* concerns a wide range of conditions that both students and teachers would like to see, but are not compulsory. Though many other conditions regarding this aspect may be considered, here we shall mention no more than two:

Gaps between teaching periods in each day must be avoided, for the sake of students and teachers alike, as well as the risk of spreading lessons over the week;

The choice of each teacher's free periods should, whenever possible, be respected.

Albeit important, these features do not cover everything one should consider when creating timetables for a Portuguese secondary school. Our intention is simply to show one way of incorporating such constraints in a mathematical model (section 4). Naturally other specific constraints may also be tackled in a similar manner.

4 — Formulation

The formulation to be presented for the STP lies within the context of nonlinear optimization with binary variables. Following the previous presentation of the STP we will enforce level 1 constraints by the model constraints, whereas level 2 constraints will be embedded through penalizations in the objective function.

Let us begin by defining the model parameters:

s — index for a room, $s = 1,\ldots,S$, where $S \geq 1$ is the number of rooms available;

d — index for a discipline, $d = 1,\ldots,D$, where $D \geq 1$ is the number of disciplines built in stage 3 of the overall timetabling process;

t — index for a time slot, $t = 1,\ldots,T$, where $T \geq 1$ is the number of time slots per week and the week is split into a number of days less than T, each containing a specified lunch period;

p — index for a teacher, $p = 1,\ldots,P$, where $P \geq 1$ is the number of teachers;

c — index for a class, $c = 1,...,C$, where $C \geq 1$ is the number of classes;

M_d — maximum number of daily time slots or lessons assigned for discipline d (a lesson occupies one time slot), $M_d \geq 0$ and integer, for $d = 1,...,D$;

N_d — number of time slots per week of discipline d, which is determined by the curriculum for the respective group of students and the work-load of the respective teacher, $N_d \geq 0$ and integer, for $d = 1,...,D$;

$discP(p)$ — set of disciplines given by teacher p, for $p = 1,...,P$;

$discC(c)$ — set of disciplines attended by students of class c, for $c = 1,...,C$;

$I_{dd'}$ — equal to 1, if disciplines d and d' are incompatible (taught by the same teacher or attended by the same group of students), and equal to 0 otherwise, for $d = 1,...,D$ and $d' = 1,...,D$;

C_{sd} — equal to 1 if room s is not suited for discipline d, and equal to 0 if otherwise, for $s = 1,...,S$ and $d = 1,...,D$;

$B_{tt'}$ — subjective penalty resulting from the occupation of time slots t and t', $B_{tt'} \geq 0$ for $t = 1,...,T$ and $t' = 1,...,T$;

G_{pt} — relative availability of teacher p in time slot t, $G_{pt} \geq 0$ for $p = 1,...,P$ and $t = 1,...,T$;

α, β, γ — penalties associated with each component of the objective function, $\alpha \geq 0$, $\beta \geq 0$ and $\gamma \geq 0$.

The following variables will be used:

x_{sdt} — equal to 1, if room s is occupied with discipline d in time slot t, otherwise equal to 0, for $s = 1,...,S$, $d = 1,...,D$ and $t = 1,...,T$.

Let us now present the objective function:

$$\text{minimize } z = \alpha\, f_1 + \beta\, f_2 + \gamma\, f_3 \qquad \textbf{(0)}$$

where:

$$f_1 = \sum_{p=1}^{P} \sum_{t=1}^{T} \sum_{t'=1,t'\neq t}^{T} B_{tt'+} \left(\sum_{s=1}^{S} \sum_{d \in discP(p)} x_{sdt}\right) \left(\sum_{s=1}^{S} \sum_{d \in discP(p)} x_{sdt'}\right);$$

$$f_2 = \sum_{c=1}^{C} \sum_{t=1}^{T} \sum_{t'=1,t'\neq t}^{T} B_{tt'} \left(\sum_{s=1}^{S} \sum_{d \in discC(c)} x_{sdt}\right) \left(\sum_{s=1}^{S} \sum_{d \in discC(c)} x_{sdt'}\right);$$

$$f_3 = \sum_{p=1}^{P} \sum_{t=1}^{T} G_{pt} \left(\sum_{s=1}^{S} \sum_{d \in discP(p)} x_{sdt}\right).$$

This objective function penalizes three aspects of the solution, all of which belonging to hierarchical level 2 for improvement — reducing gaps and time-spreading, both for teachers and for students and imposing preferences for teachers. By changing the respective parameters we can change the relative importance of these aspects.

The first and second components, f_1 and f_2, impose timetables for teachers and students respectively, with a low spread factor. This low spread factor refers to the lowest possible number of free periods per day and avoidance of lessons spread over the week.

The third part of the objective function, f_3, incorporates teacher preferences in the model.

Optimization of the above function is performed, while respecting the constraints explained below.

This first set of constraints determines maximum occupation of each room per discipline at each time slot.

$$\sum_{d=1}^{D} x_{sdt} \leq 1 \quad (s = 1,...,S;\ t = 1,...,T) \qquad \textbf{(1)}$$

For each teacher, constraints **(2)** impose a maximum of one lesson per time slot.

$$\sum_{s=1}^{S} \sum_{d \in discP(p)} x_{sdt} \leq 1 \quad (p = 1,...,P;\ t = 1,...,T) \qquad \textbf{(2)}$$

Each constraint **(3)** restricts the number of lessons for discipline d per week, as indicated by the curriculum of the group of students relative to discipline *d*.

$$\sum_{s=1}^{S} \sum_{t=1}^{T} x_{sdt} = N_d \quad (d = 1,...,D) \qquad \textbf{(3)}$$

Constraints **(4)** respect two different aims. The first one is to prevent the time clashes for each teacher already imposed by **(2).** This redundancy plays an important role for the neural network formulation [see reference **(5)**]. The aim of the second is to exclude the concurrent teaching of disciplines which are common to the same group of students:

$$\sum_{d=1}^{D} \sum_{d' \neq d} I_{dd'} \sum_{s=1}^{S} x_{sdt} \sum_{s'=1}^{S} x_{s'd't} = 0 \quad (t = 1,...,T) \qquad \textbf{(4)}$$

Next constraints **(5)** impose suitable rooms for each discipline in each time slot.

$$\sum_{d=1}^{D} \sum_{s=1}^{S} C_{sd}\ x_{sdt} = 0 \quad (t = 1,...,D) \qquad \textbf{(5)}$$

For reasons of simplification, but without limiting the general nature of the present formulation, the following three sets of constraints will be drawn up for the specific case of T= 50, in other words 10 daily time slots in a five-day week, as illustrated in table 1. It is also considered without limitations that the lunch period is one of the two time slots between 12 a. m. and 2 p. m.

TABLE 1

Daily time slots

Hours	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-9:00	1	11	21	31	41
9:00-10:00	2	12	22	32	42
10:00-11:00	3	13	23	33	43
11:00-12:00	4	14	24	34	44
12:00-13:00	5	15	25	35	45
13:00-14:00	6	16	26	36	46
14:00-15:00	7	17	27	37	47
15:00-16:00	8	18	28	38	48
16:00-17:00	9	19	29	39	49
17:00-18:00	10	20	30	40	50

The maximum number of lessons for each discipline per day is bound by imposing constraints **(6)**:

$$\sum_{s=1}^{S} \sum_{t=h+1}^{h+10} x_{sdt} \leq M_d \quad (d=1,...,D;\ h=0,10,20,30,40) \qquad \textbf{(6)}$$

Both teachers [constraints **(7)**] and students [constraints **(8)**] must have at least one free hour within the pre-defined lunch period:

$$\sum_{d \in discP(p)} \sum_{s=1}^{S} \sum_{t=h+1}^{h+2} x_{sdt} \leq 1 \quad (p=1,...,P;\ h=4,14,24,34,44) \qquad \textbf{(7)}$$

$$\sum_{d \in discC(c)} \sum_{s=1}^{S} \sum_{t=h+1}^{h+2} x_{sdt} \leq 1 \quad (c=1,...,C;\ h=4,14,24,34,44) \qquad \textbf{(8)}$$

Lastly we come to the binary conditions for the variables:

$$x_{sdt} = 0,1(s=1,...,S;\ d=1,...,D;\ t=1,...,T). \qquad \textbf{(9)}$$

The above formulation is, in fact, very general and can easily be adapted for almost all cases of class timetabling — even the more complicated ones. Though ill-suited for exact problem solution methodologies it can be the basis of a heuristic method to tackle the problem [Carrasco (1996)]. Let us show how the formulation works with an instance of STP.

5 — The case of a small-sized school

The present section refers to a small instance of the STP, obtained from a hypothetical small-sized school with the following characteristics:

A week's period of 3 days, each with 3 time slots;
4 subjects (Mathematics, Portuguese, Geography and English);

6 teachers (Teacher 1,..., Teacher 6);
4 classes (Class A, Class B, Class C and Class D);
3 rooms (1 large, 1 medium sized and 1 small).

The 9 time slots were allocated according to table 2.

TABLE 2

Daily time slots for the example

Hours	Day 1	Day 2	Day 3
9:00-12:00	1	4	7
13:00-16:00	2	5	8
16:00-19:00	3	6	9

Here (table 3) we have the curriculum for each class and the assignment of teachers to classes.

TABLE 3

Curricula and assignment of teachers for the example

Subjects	Teachers	Class A	Class B	Class C	Class D	Number of lessons
Mathematics	Teacher 1	2		3		5
Mathematics	Teacher 2		2		3	5
Geography	Teacher 3	2	2	1		5
Portuguese	Teacher 4	2	2			4
Portuguese	Teacher 5			2	3	5
English (option)	Teacher 6	1				1
						25

It should be mentioned that the contents of the above table provide the values for the N_d parameter of the formulation we are studying. Moreover, the table defines the 12 disciplines for this instance: Math 1A (Mathematics for Class A with Teacher 1), Math 1C, Math 2B, Math 2D, Geo 3A, Geo 3B, Geo 3C, Port 4A, Port 4B, Port 5B, Port 5C and Engl 6AB. They will be rated in the above order, so that $d = 1$ will stand for Math 1A, $d = 2$ for Math 1C, and so on.

It should be stressed that the maximum number of teaching slots or lessons is 27 (3 rooms × 9 time slots), irrespective of room capacity and suitability. On the other hand, the curriculum requires 25 lessons for classes per week. Consequently, the current timetabling issue is almost saturated, as almost all possible teaching slots will be occupied by classes in one «week» — an *occupation rate* of 93%. Obviously real problems are not so saturated as this exemplar.

The matrix given in table 4 expresses the clashes between disciplines attended by the same group of students and among disciplines of the same teacher. That is, if discipline d and discipline d' belong to the same teacher, then $I_{dd'} = 1$, and also $I_{dd'} = 1$ if disciplines d and d' share the same group of students. $I_{dd'} = 0$ if there are no clashes between disciplines d and d'.

TABLE 4

Incompatibility between disciplines for the example

Disciplines	Math1A	Math1C	Math2B	Math2D	Geo3A	Geo3B	Geo3C	Port4A	Port4B	Port5C	Port5D	Engl6AB
Math 1A	-	1			1			1				1
Math 1C	1	-					1			1		
Math 2B			-	1		1			1			1
Math 2D			(*)1	-							1	
Geo 3A	1				-	1	1	1				1
Geo 3B			1		1	-	1		1			1
Geo 3C		1			1	1	-			1		
Port 4A	1				1			-	1			1
Port 4B			1			1		1	-			1
Port 5C		1					1			-	1	
Port 5D				1						1	-	
Engl 6AB	1		1		(**)1	1		1	1			-

Let us, for instance, examine the cells marked with (*) and (**). The first one, (*), results from the fact that Teacher 2 delivers both disciplines Math 2D and Math 2B. The second mark, (**), arises from the fact that disciplines Engl 6AB and Geo 3A cannot be assigned to the same time slot as they are attended by the same group of students from Class A.

Table 5 displays the matrix $[C_{sd}]_{s=1,...S,d=1,...,D}$ which stipulates specific rooms for each discipline. In the last column this table gives the daily bounds in the number of lessons for each discipline, in other words the value M_d. This matrix shows a *rate of room/discipline incompatibility* of about 42%.

TABLE 5

Room suitability and maximum number of lessons per discipline for the example

Disciplines	Room adequacy			Maximum number of daily lessons
	Room 1	Room 2	Room 3	
Math 1A	0	0	1	1
Math 1C	0	0	1	2
Math 2B	0	0	1	2
Math 2D	0	0	1	2
Geo 3A	0	1	1	1
Geo 3B	0	1	1	2
Geo 3C	0	1	1	1
Port 4A	0	1	0	2
Port 4B	0	1	0	2
Port 5C	0	1	0	1
Port 5D	0	1	0	3
Engl 6AB	0	1	0	1

In the present case, due to the small number of time slots per day (only 3), there are no lunch break constraints. Instead, we inserted 1 compulsory hour for lunch, 12:00 to 13:00.

Table 6 illustrates the teachers' preferences.

TABLE 6

Teachers' preferences for the example

Hours	Day 1	Day 2	Day 3
Teacher 1			
9:00-12:00	2	1	1
13:00-16:00	2	1	1
16:00-19:00	2	1	1
Teacher 2			
9:00-12:00	1	1	2
13:00-16:00	1	1	2
16:00-19:00	1	1	2
Teacher 3			
9:00-12:00	2	1	1
13:00-16:00	2	1	1
16:00-19:00	2	1	1
Teacher 4			
9:00-12:00	1	2	1
13:00-16:00	1	2	1
16:00-19:00	1	2	1
Teacher 5			
9:00-12:00	1	1	2
13:00-16:00	1	1	2
16:00-19:00	1	1	2
Teacher 6			
9:00-12:00	2	1	1
13:00-16:00	2	1	1
16:00-19:00	2	1	1

Finally, table 7 shows the values for matrix $[B_{tt'}]_{t,t'=1,\ldots T}$ penalizing the spread of teaching slots per day and per «week».

TABLE 7

Penalization of free periods for the example

T. slots	1	2	3	4	5	6	7	8	9
1	0	0	1	2	2	2	3	3	3
2	0	0	0	2	2	2	3	3	3
3	1	0	0	2	2	2	3	3	3
4	2	2	2	0	0	1	2	2	2
5	2	2	2	0	0	0	2	2	2
6	2	2	2	1	0	0	2	2	2

T. slots 1	2	3	4	5	6	7	8	9	
7	3	3	3	2	2	2	0	0	1
8	3	3	3	2	2	2	0	0	0
9	3	3	3	2	2	2	1	0	0

Though corresponding to an unrealistic small school, this model requires 324 binary variables and more than one hundred constraints.

For illustrative purposes, let us now present a feasible solution for this STP instance obtained through a heuristic procedure found in Carrasco (1996). The variables with non-zero values are as follows:

$x_{182}=x_{153}=x_{134}=x_{175}=x_{146}=x_{167}=x_{168}=x_{159}=1$ (lessons in Room 1);
$x_{241}=x_{242}=x_{214}=x_{235}=x_{226}=x_{217}=x_{228}=x_{229}=1$ (lessons in Room 2);
$x_{391}=x_{3,10,2}=x_{3,11,3}=x_{3,11,4}=x_{3,11,5}=x_{3,12,6}=x_{3,10,7}=x_{388}=x_{399}=1$ (lessons in Room 3).

If we convert their values into scheduling form, timetables are created both for teachers and for students, as illustrated in table 8. Here, as there are no different optional subjects within each class, the timetables for all students of each class are identical.

TABLE 8

Timetables for teachers in the example

Hours	Day 1	Day 2	Day 3
	Teacher 1		
9:00-12:00	******	Math A - Room 2	Math A - Room 2
13:00-16:00	******	—	Math C - Room 2
16:00-19:00	******	Math C - Room 2	Math C - Room 2
	Teacher 2		
9:00-12:00	Math D - Room 2	Math B - Room 1	******
13:00-16:00	Math D - Room 2	Math B - Room 2	******
16:00-19:00	—	Math D - Room 1	******
	Teacher 3		
9:00-12:00	******	—	Geo B - Room 1
13:00-16:00	******	Geo C - Room 1	Geo B - Room 1
16:00-19:00	***Geo A - Room 1***	—	Geo A - Room 1
	Teacher 4		
9:00-12:00	Port B - Room 3	******	—
13:00-16:00	Port A - Room 1	******	Port A - Room 3
16:00-19:00	—	******	Port B - Room 3
	Teacher 5		
9:00-12:00	—	Port D - Room 3	***Port C - Room 3***
13:00-16:00	Port C - Room 3	Port D - Room 3	******
16:00-19:00	Port D - Room 3	—	******

TABLE 8

Timetables for teachers in the example

Hours	Day 1	Day 2	Day 3
	Teacher 6		
9:00-12:00	******	—	—
13:00-16:00	******	—	—
16:00-19:00	******	Engl A+B - Room 3	—

TABLE 9

Timetables for students (classes) in the example

Hours	Day 1	Day 2	Day 3
	Class A		
9:00-12:00	—	Math - Room 2	Math - Room 2
13:00-16:00	Port - Room 1	—	Port - Room 3
16:00-19:00	Geo - Room 1	Engl - Room 3	Geo - Room 1
	Class B		
9:00-12:00	Port - Room 3	Math - Room 1	Geo - Room 1
13:00-16:00	—	Math - Room 2	Geo - Room 1
16:00-19:00	—	Engl - Room 3	Port - Room 3
	Class C		
9:00-12:00	—	—	Port - Room 3
13:00-16:00	Port - Room 3	Geo - Room 1	Math - Room 2
16:00-19:00	—	Math - Room 2	Math - Room 2
	Class D		
9:00-12:00	Math - Room 2	Port - Room 3	—
13:00-16:00	Math - Room 2	Port - Room 3	—
16:00-19:00	Port - Room 3	Math - Room 1	—

It should be noted that the above schedules are of reasonable quality. There are no more than two gaps in the timetable of Class A and Teacher 1. As for teachers' preferences for free specific slot times, the above timetables respect all but two cases (see timetables for Teacher 3 and Teacher 5).

6 — Closing comments

The problem of creating timetables at a secondary school addressed in this paper was formulated as a binary nonlinear problem. The reason for choosing a binary nonlinear model was largely due to the fact that our problem is not suited to decomposition into small problems. If such were the case, we could apply other alternative optimization models, such as the more attractive network based models.

The STP is a difficult problem and may easily reach considerable dimensions. Even if we study the timetabling problem of a school broken down into cycles (grades 7-9, 10-11 and 12) and shifts (morning, afternoon and evening) the number of variables may total about 200 thousands in a medium-sized school with 100 teachers, 40 rooms and 2500 students.

In view of these caracteristics, our problem is not suited to classic exact solving methods directly based on this huge formulation, but rather to lighter methodologies like local search modern strategies embedding tabu constraints, genetic evolution and annealing schemes [Reeves (1993)].

However, Carrasco (1996) decided to heuristically tackle the timetable issue by using this precise formulation through a Hopfield neural network. In fact, the highly constrained assignment formalization for the STP seemed appropriate to neural network based heuristic techniques. This methodology, together with its application to pseudo-real situations, is presented in a companion paper [Carrasco and Pato (1997)]. The results obtained so far have encouraged us to tackle a real life problem, by refining the neural network based algorithm.

7 — References

[1] D. Abramson (1991), «Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms», *Management Science*, vol. 37, pp. 83-113.

[2] D. Abramson and Abela (1991), «A Parallel Genetic Algorithm for Solving the School Timetabling Problem», Technical Report, Division of I.T., C.S.I.R.O., Royal Melbourne Institute of Technology, Melbourne.

[3] R. Alvarez-Valdes, G. Martin and J. M. Tamarit (1995), «Constructing Good Solutions for a School Timetabling Problem», Technical Report, Department of Statistics and Operations Research, University of Valencia.

[4] M. P. Carrasco (1996), Redes Neuronais na «Elaboração de Horários Escolares, Master Thesis», Instituto Superior de Economia e Gestão, UTL, Lisboa, submitted for publication.

[5] M. P. Carrasco and M. V. Pato (1997), «Um Modelo Neuronal para a Elaboração de Horários Escolares», aceite para publicação em *Investigação Operacional*.

[6] M. W. Carter (1995), «Timetabling Literature», available from *ftp.cs.nott.ac.uk* in directory *ttp*.

[7] M. W. Carter, G. Laporte and J. W. Chinneck (1994), «A General Examination Scheduling System», *Interfaces*, vol. 24, pp. 52-58.

[8] A. Colorni, M. Dorigo and V. Maniezzo (1990), «Genetic Algorithms and Highly Constrained Problems: The Timetable Case», *Proceedings of the First International Conference on Parallel Problem Solving from Nature*, pp. 55-59.

[9] T. B. Cooper and J. H. Kingston (1995), «The Complexity of Timetable Construction Problems», Technical Report, Basser Department of Computer Science, The University of Sydney, Sydney.

[10] D. Costa (1994), «A Tabu Search Algorithm for Computing an Operational Timetable», *European Journal of Operational Research*, vol. 76, pp. 98--110.

[11] K. A. Downsland (1990), «A Timetabling Problem in which Clashes are Inevitable», *Journal of the Operational Research Society*, vol. 41, pp. 907--918.

[12] R. W. Eglise and G. K. Rand (1987), «Conference Seminar Timetabling», *Journal of the Operational Research Society*, vol. 38, pp. 591-598.

[13] S. Even and A. Shamir (1976), «On the Complexity of Timetable and Multicommodity Flow Problems», *SIAM Journal of Computing*, vol. 5, pp. 691-703.

[14] J. A. Ferland and S. Roy (1985), «Timetabling Problem for University as Assignment of Activities to Resources», *Computers and Operations Research*, vol. 12, pp. 207-218.

[15] L. Gislén, C. Peterson and B. Soderberg (1989), «'Teachers and Classes' with Neural Networks», *International Journal of Neural Systems*, vol. 1, pp. 167-176.

[16] L. Gislén, C. Peterson and B. Soderberg (1992), «Complex Scheduling with Potts Neural Networks», *Neural Computation*, vol. 4, pp. 805-831.

[17] A. Hertz (1991), «Tabu Search for Timetabling Problems», *European Journal of Operational Research*, vol. 54, pp. 39-47.

[18] D. Jonhson (1990), «Timetabling University Examinations», *Journal of the Operational Research Society*, vol. 41, pp. 39-47.

[19] G. Laporte and S. Desroches (1984), «Examination Timetabling by Computer», *Computers and Operations Research*, vol. 11, pp. 351-360.

[20] C. R. Reeves (1993), *Modern Heuristic Techniques for Combinatorial Problems*, editor, Backwell Scientific Publications, Oxford, England.

[21] D. de Werra (1985), «An Introduction to Timetabling», *European Journal of Operational Research*, vol. 19, pp. 151-162.

[22] M. Write (1996), «School Timetabling Using Heuristic Search», *Journal of the Operational Research Society*, vol. 47, pp. 347-357.