MILITARY TECHNICAL COLLEGE
CAIRO - EGYPT

ICEENG 98

FIRST INTERNATIONAL CONF. ON
ELECTRICAL ENGINEERING

ANALYSIS AND MODELING OF A TIGHTLY CONSTRAINED HYBRID TIMETABLING SYSTEM

Dr. Eng. M. Samy Gamal El Din*

Dr. Eng. Ismail A. Farag*

Eng. Khaled Elshwadfy*

Dr. Eng. Kamel El Hadad*

Dr. Eng. A. Hasaballa Khalil*

Abstract

This paper presents the analysis and modeling of a special class of the timetabling problem. The introduced problem is a hybrid one. It combines the school and the university timetabling cases. The set of imposed constraints on the introduced problem is a tight one. This class of problems can be identified in some institutions such as Military Technical College.

1. Introduction

The course timetabling is the process of scheduling a sequence of meetings between instructors and students in a prefixed period of time (typically a week), satisfying a set of constraints of various types. A large number of variants of the timetabling problem have been proposed in the literature, which differ from each other, based on the type of institution involved (university or high school) and the type of constraints. The problem has been traditionally considered in the operation research field. It has recently been tackled with techniques that belong to artificial intelligence field (e.g. genetic algorithms, Tabu search, simulated annealing and constraint satisfaction).

This paper studies the modeling of a hybrid class of course timetabling problem. This class of problem combines both the school and university timetabling. This class of problem can be identified in some institutions. The Military Technical College is one of these institutions. The imposed set of constraints in such institution is a very tight one.

2. The General Timetabling Problem

2.1 Formal Definitions

A **timetable**, or schedule, is a description of the movement and grouping of resources over time, often to achieve a certain aim or aims and/or subject to a set of constrains [10]. Resources may cover any number of objects which may be grouped according to type, for example, a type of resource in a school timetable would be a class and a specific instance of the type: Form 10B Similarly, we could have trains, machines, instructors, rooms and slide projectors or anything else that may be required.

The definition of the timetabling problem is found in literature in many forms. The following definitions represent the most famous ones:

Timetabling problem is the assignment of time slots for the courses afford in semester so that the various requirements and constraints are satisfied [17]. A time slot is defined by the day of the week and time period. The period beginning time and duration describe a period.

While another definition of the **timetable** is as it centrally involves the need to assign times to a set of events, subject to constraints on these assignments, simply a pair of events must not overlap in time. This may be because, for example the two events are lectures and one or more students must sit both of them, or they may be classes given by the same instructor, and so on [14].

Also the **Timetabling problem** could be defined as the problem of creating a valid timetable involves scheduling classes, instructors and rooms into a fixed number of periods, in such a way that no instructor, class or room is used more than once per period. A particular combination of an instructor, a subject, a room and a class is called tuple. A tuple may be required more than once per week. Thus, the timetabling problem can be phrased as scheduling a number of tuples such that an instructor, class or room does not appear more than once per period, [1].

* Staff Members of the Military Technical College

Where, [11] introduces the problem in its simplest form:

Timetabling is the process of assigning courses or exams to periods and rooms such that either no conflicts or minimum number of conflicts occur. By "conflicts", we mean when students or staff are scheduled to be in the same place or when a room is scheduled to be used by more than one group at the same time. [11].

At last, [20] defines the **Timetabling problem** as assigning times and places to a many separate lectures, tutorials, ...etc. to satisfy several constraints concerning capacities and locations of available rooms. Free-time needs and other such considerations for lecturers, and relationships between particular courses.

A **feasible timetable**, or schedule, is one that satisfies its associated set of constrains [10]. Such constraints might include: groupings that must occur at some point; groupings that may never occur; that if one resource is present, so must another be; that one required grouping occurs before another; that all groupings must involve a certain amount of each type of resource.

The most prominent overall constraint (central to all timetabling problems) is that there should be no clashes; that is any pair of lectures which are expected to share common students or instructors should not be scheduled simultaneously. [20].

2.2 Specific Timetables

Each different occurrence of the general scheduling problem carries with it its own sets of jargon, rules and requirements. Very often, the problems are so different that they may hardly be classified as of the same form, and more often that not an efficient solution methodology for one will prove inefficient on another. A few of the more common occurrences of the timetabling or scheduling problem are described briefly below.

2.2.1 School Timetables

The school timetable describes when each class has a particular lesson and in which room it is to be held. The actual content of the timetable is largely driven by the *curriculum*; the number of hours of each subject taught per week is often set nationally. Each class consists of a set of pupils, who must be occupied from the time they arrive until the time they leave school, a specific instructor being responsible for the class in any one period [10].

Instructors are usually allocated in advance of the timetabling process, so the problem is to match up meetings of instructors with classes to particular time slots so that each particular instructor meets every class he or she is required to. Obviously each class or instructor may not be involved in more than one meeting at a time. Often, it is required that each instructor has at least one morning or afternoon free per week. Many other similar constraints may exist.

2.2.2 Rail Timetables

The rail timetables describes the movement of trains around the country, when they should leave a station and when they are due to arrive at the next one. Unlike school timetables that are specified on a period by period basis, this schedule must be defined to the minute or even smaller time unit.

Clearly, restrictions exist in terms of the track and surrounding infrastructure. The frequency of signals dictates how often a train may use a piece of track. The presentation of the timetable is dependent on the information required by the user. Someone wishing to travel from Cairo to Aswan would not normally be interested in local trains along the Nile. The size of the stations (and the number of tracks) dictates a maximum number of trains in any one place at once. Staffing levels also present similar constraints: each train might require a driver and at least one guard.

Certain qualities may be ascribed to a good timetable. Is the timetable robust? If a train breaks down, how long before a replacement can be found? What proportion of train journeys are passenger carrying and what proportion are not? Is the timetable likely to cause two trains to be on the same track at once? If satisfactory answers cannot be found to these questions, then the timetable designers might ask themselves do we need more trains or should we discontinue services? [10]

2.2.3 Job Shop Schedules

Perhaps the most researched and most quoted instance of the scheduling problem is Job Shop Scheduling. On a factory floor are a number of machines. The factory has a number of orders, which must be processed, each generally requiring the use of several machines, in a particular order. Some machines might process one item at once; some might take any number of items at the same time. The problem is further complicated in that new orders arrive all the time whilst old orders are finished and packed off to the customer.

Even a simple situation, such as in Fig-1 can produce conflict. How often is *machine2* to be run? Do we maximize efficiency by running it only once *machine1* has processed n times, or do we maximize speed of output by running it each time it receives an input? [10]

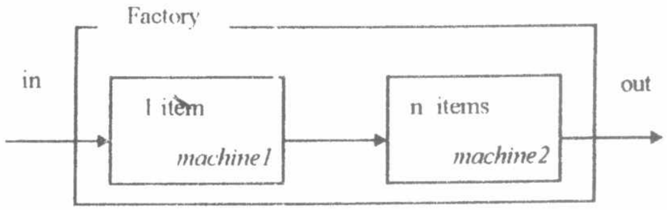


Fig-1 *A simple Job-Shop Scheduling Problem*

2.2.4 University Course Timetable

Although, in principal, much likes a school table, the construction of a University Course Timetable is quite different. Many different departments each offer a multitude of courses from which students are required to take some and choose a number of other (this way is used in most of the universities around the world) [10], [21]. In some cases, each department is responsible for its own timetable and must try to take into the account the timetables of other departments. Where, in other cases, the whole timetable is decided upon centrally.

The main difference with the high school problem is that university courses can have common students, whereas school classes are disjoint sets of students. If two courses have common students then they conflict, and they cannot be scheduled at the same period. Moreover, schoolteachers always teach to more than one class, whereas in universities, a professor may teach only one course. In addition, in the university problem, availability of rooms (and their size) plays an important role, whereas in high school problem they are often neglected because, in most cases, we can assume that each class has its own room.

2.3 The Complexity of Timetabling

There are several reasons why we might wish to know how difficult a particular problem is to solve. From the point of view of a developer of timetabling software, it would help in the generation of test problems comparable with those that must actually be solved and would help in the comparison of different methodologies. From the point of view of someone wishing to select a particular approach for their own problem, it would give a guide as to what sort of method is required and what sort of solution may be produced.

In some cases, the timetabling problem consists of finding any timetable that satisfies all the constraints. In this case, the problem is formulated as a ***search problem***. In other cases, the problem is formulated as an ***optimization problem***, that is, the required is a timetable that satisfies all the hard constraints and minimizes (or maximizes) a given *objective function* which embeds the soft constraints. In some approaches, the optimization formulation is just a means to apply optimization techniques to a search problem.

In both cases (search and optimization), we define the underlying problem, which is the problem of deciding if there exists a solution with a given value of the objective function, in the case of an optimization problem. When we mention the complexity of the problem, we refer to the complexity of the underlying decision problem. Therefore, an exact solution is achievable only for small cases (less than 10 courses), whereas real instances usually may involve a few hundreds of courses. It follows that heuristic methods are feasible, which do not guarantee to reach the (optimal) solution. The artificial intelligence techniques can modify the computational aspects of a given problem, [11], [21].

[10] gives some advice on how to select an appropriate timetabling heuristic, suggesting that one that is known to be effective is tried, concentrating solely on producing a conflict-free timetable, and then (if possible), extra constraints can be incorporated. This provides a simple assessment of how difficult the timetabling problem is. To produce an effective measure, however, it would be more appropriate to use a random based algorithm over several runs since examples may be constructed where deterministic algorithms performs very badly.

To find the difficulty of the problem, clearly, the heuristic (or random) solution must be compared with the requirements of the institution. The smaller the amount of time available in comparison to the amount of time used by the heuristic algorithm, the harder the timetable will be to find.

[11] mentioned that timetabling and constraint satisfaction are very hard problems and belong to the ***NP-Complete*** class of problems for which a general polynomial time deterministic algorithm is not known. In the timetabling problem, the process of finding a period for each tuple (course, instructor, room or class) has been shown to be equivalent to assigning colors to vertices in a graph so that no adjacent vertices always have different colors, [10].

[10] presented computational complexity results relating to a range of difficult problems including the optimal *coloring of random graphs*. These problems he grouped under the heading of *NP-complete*. These are the problems that may not be solved in polynomial time using deterministic algorithm. No deterministic polynomial time algorithm is known for any of these problems and most people believe that such an algorithm does not exist. This result implies that the use of an exact algorithm to find an

optimal coloring of the timetable graph would be too time consuming to be practical and that instead good approximate methods must be found. In fact, the problem of finding a k coloring (a coloring using exactly k colors) is also NP-complete.

2.4 Timetables Quality

Targets for timetable quality cannot be set before a solution has been attempted, at least not without an effective measure of the problem's difficulty. This is because the quality of the solution is completely dependent on the difficulty of the problem and the efficiency of method used to produce the timetable. In general, the only requirement is to compare timetables rather than create benchmark since it is inconceivable that any institution, with a complex timetabling problem, should not already have an exception of the quality of possible timetables given previous solutions of its own problem.

Generally, timetable quality is very subjective and its definition changes from problem to problem. The usual measure, which is common to all applications, is in terms of the number of constrains that are broken, [10].

Two important measures of timetable quality relate specifically to its feasibility and arise directly from the graph-coloring problem. These are the requirements that no student is required to sit more than two periods with the same course sequentially and that there must be sufficient seats available for the number of students scheduled. As has been said, a timetable that satisfies these constraints is considered feasible but this does not necessarily mean that it is sufficiently good to be used by an institution.

In terms of producing acceptable or good timetables one thing should be born in mind above all others: "As Lions observed 25 years ago the ultimate test of a computer generated timetable is ***Will the school or the college buy it?*** ". The concept of an optimal timetable is hard to define. As mentioned above, it seems to perceive a difference between what is called the optimal timetable and the one which staff would accept. This begs the question '*If the staff will not accept the timetable, how can it be optimal'?* The most important factor here is whether staff accepts the timetable or not. As [10] states "There is no need to seek an 'optimal' solution, in fact in the context of college timetables this concept is hard to define." It is necessary, though, to agree on one or more measures of quality as otherwise Table-1, it would be impossible to produce quality timetables.

- Lectures should be scheduled early in the timetable
- Some courses may only take place in certain rooms
- Courses should be located near to their home departments
- Some courses should be scheduled in a particular order

Table-1 *Some Examp le Timetabling Quality Measures*

We should have a numerical value of quality for each constraint; there are two ways in which we can distinguish between timetables. The weighted penalty function approach, involves giving each timetable one single quality rating calculated as follows

$$\textit{Timetable Score} = a_1c_1 + a_2c_2 + a_3c_3 + \ldots + a_nc_m$$

Where $a_1, a_2, \ldots a_n$ are weights quantifying the importance of each constraint and $c_1, c_2, \ldots, c_n$ are actual numerical stores for each constraint satisfied. The timetable with the highest score is then selected as the one to be used.

Another method [10], which involves keeping a store of potential timetables, is to keep a vector of values and then keep those that are not dominated by any others i.e. are not outperformed in every component. This is called the set of Pareto optimal timetables.

$$\overline{\textit{Timetable Score}} = \{C_1\ C_2, C_3, \ldots, C_m\}$$

The Institution may then take its own decision as to which is to be used.

3. Problem Formulation

In this section, we describe in detail the high school timetabling problem, also known as *class instructor model*. We start describing a simplified version, which can be solved in polynomial time; therefore, we move to the basic formulation. Although that is still not a "real" problem, it has the minimal set of constraints that makes it a hard problem, and its solution requires the heuristic techniques employed also for the more complex cases. Subsequently, we introduce the corresponding optimization problem, and describe some variants of it considered in the literature. Finally, we discuss solution techniques and approaches.

Simplified polynomial problem

We have m classes $c_1, \ldots, c_m$, n teachers $t_1, \ldots, t_n$, and p periods $1, \ldots, p$. We are also given a non-negative integer matrix $R_{m \times n}$, called *Requirements matrix*, where r_{ij} is the number of lectures given by teacher t_j to class c_i.

The problem consists in assigning lectures to periods in such a way that no teacher or class is involved in more than one lecture at a time. The mathematical formulation is as follows [21]:

<u>**TTP1**</u>

Find x_{ijk} $(i = 1, \ldots, m; \quad j = 1, \ldots, n; \quad k = 1, \ldots, p)$

St. $$\sum_{k=1}^{p} x_{ijk} = r_{ij} \qquad (i = 1,\ldots,m; \quad j = 1,\ldots,n) \tag{1}$$

$$\sum_{j=1}^{n} x_{ijk} \le 1 \quad (i = 1,\ldots,m; \quad k = 1,\ldots,p) \tag{2}$$

$$\sum_{i=1}^{m} x_{ijk} \le 1 \quad (j = 1,\ldots,n; \quad k = 1,\ldots,p) \tag{3}$$

$$x_{ijk} = 0 \; or \; 1 \quad (i = 1,\ldots,m; \quad j = 1,\ldots,n; \quad k = 1,\ldots,p) \tag{4}$$

Where $x_{ijk} = 1$ if class c_i and teacher t_j meet at period k, and $x_{ijk} = 0$ otherwise.

Constraint (1) ensure that each teacher gives the right number of lectures to each class. Constraint (2) (respectively Constraint (3)) ensure that each teacher (respectively class) is involved in at most one lecture for each period.

In addition, [21] proves that there exists always a solution of this problem, unless a teacher or a class is required to be involved in more than p lectures. More precisely, there exists a solution if and only if

$$\sum_{i=1}^{m} r_{ij} \le p \qquad (j = 1,\ldots,n) \tag{5}$$

$$\sum_{j=1}^{n} r_{ij} \le p \qquad (i = 1,\ldots,m) \tag{6}$$

In order to solve TTP1, we may associate to an instance of the problem a bipartite multi-graph: Classes and teachers are associated to vertices, and each class c_i is linked to each teacher t_j by r_{ij} parallel edges. The solution technique is based on finding a sequence of maximal matching in the resulting bipartite multi-graph, where a matching is a set of edges with no common nodes.

Alternatively, the problem can be reduced to a problem of edge coloring on graphs: Given p colors (each period corresponding to a color), the problem consists of finding an assignment of a color to each edge such that no two adjacent edges have the same color. Thereafter, the variable x_{ijk} gets value 1 if one of the edges between c_i and t_j gets color k.

[21] considers also some variants TTP1 that are still solvable in polynomial time. He considers the possibility that a teacher (and a class) can be involved in more than one lecture for each period. In such variant, a period represents not an atomic time slot but a set of them (for example, a day). He also considers the case in which the lectures are constrained so that they must be spread as much as possible throughout all the periods.

Basic search problem

The problem TTP1 does not include any constraints on the possible scheduling of the lectures. In real instances, instead, we must take into account the possibility that a teacher (or a class) is unavailable at a given time.

We now introduce the school timetabling problem with unavailability of teachers and classes. [21] introduces two binary matrices T_{mxp} and C_{nxp} such that $t_{ik} = 1$ (respectively $c_{jk} = 1$) if teacher t_i (respectively class c_j) is available at period k, and $t_{jk} = 0$ (respectively $c_{jk} = 0$) otherwise. Thereafter, he replaces Constraints (2) and (3) in TTP1 by Constraints (7) and (8) as below:

TTP2

Find $$x_{ijk} \qquad (i = 1,\ldots,m; \quad j = 1,\ldots,n; \quad k = 1,\ldots,p)$$

St. $$\sum_{k=1}^{p} x_{ijk} = r_{ij} \qquad (i = 1,\ldots,m; \quad j = 1,\ldots,n) \tag{1}$$

$$\sum_{j=1}^{n} x_{ijk} \le t_{ik} \qquad (i = 1,\ldots,m; \quad k = 1,\ldots,p) \tag{7}$$

$$\sum_{i=1}^{m} x_{ijk} \le c_{jk} \qquad (j = 1,\ldots,n; \quad k = 1,\ldots,p) \tag{8}$$

$$x_{ijk} = 0 \; or \; 1 \qquad (i = 1,\ldots,m \quad j = 1,\ldots,n; \quad k = 1,\ldots,p)$$

[21] considers also constraints due to preassignments: A particular lecture can be imposed to be scheduled at a given time. Reassignments can be expressed by adding a set of constraints of the following form

$$x_{ijk} \geq p_{ijk} \qquad (i = 1,\ldots,m; \qquad j = 1,\ldots,n; \qquad k = 1,\ldots,p) \quad (9)$$

Where $p_{ijk} = 0$ if there is no preassignment, and $p_{ijk} = 1$ when a lecture of teacher t_j to class c_i is preassigned to period k. [21] also shows that unavailability can be expressed as preassignments with dummy classes or teachers.

Optimization Problem

The problem TTP2 is a search problem, whose solution is any feasible timetable. However, in real applications, a feasible timetable can be better than another one, and the goal is to find the optimal one. This consideration forces us to formulate the timetabling problem as an optimization problem with an *objective function* to minimize (or maximize). [21] proposes to add to the basic problem TTP2 the following objective function

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} d_{ijk}\, x_{ijk} \qquad (10)$$

Where a large d_{ijk} is assigned to periods k in which a lecture of teacher t_j to class c_i is less desirable.

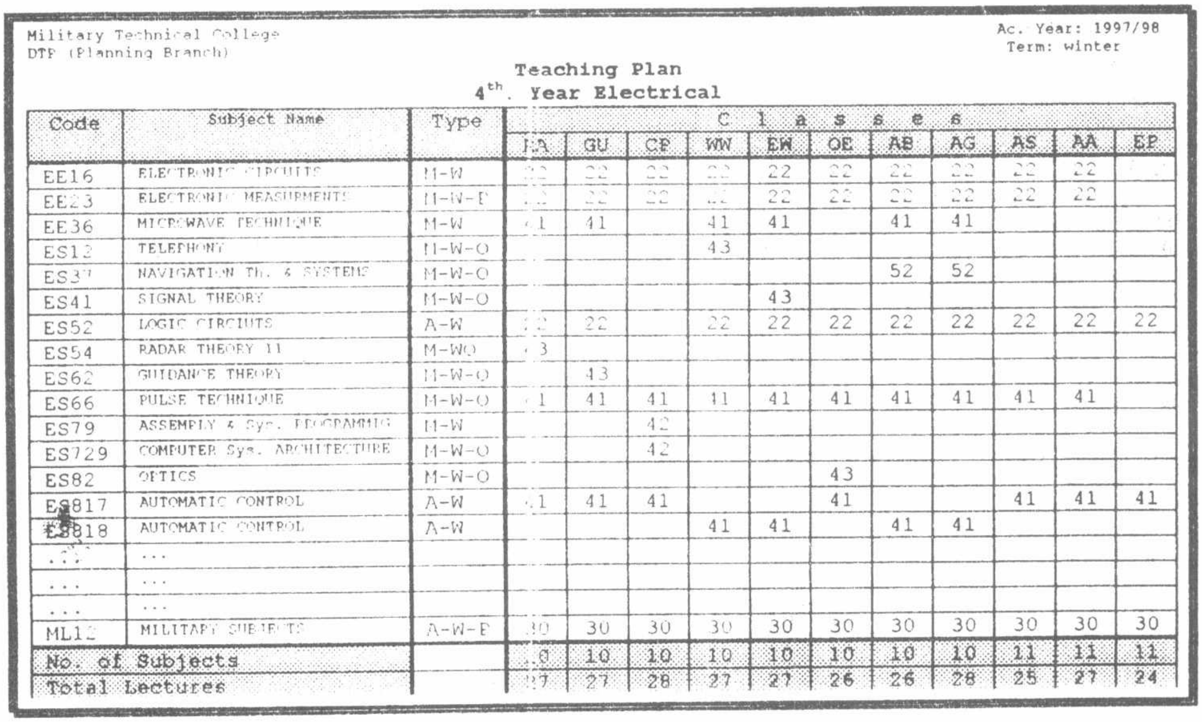
[21] introduced a more complex objective function based on the following quantities (with decreasing weight):

- The *didactic cost*: e.g. spreading the lectures over the whole week;
- The *organizational cost*: e.g. having a teacher available for possible temporary teaching posts;
- The *personal cost*: for example a specific day-off for each teacher.

He introduced a different approach based on constraint language, and associates a penalty for each constraint violated. Their objective is to minimize the overall penalty. As an example of constraint violation, they consider the possibility that a teacher is forced to teach in a period in which he is not available.

4. Analysis of a Hybrid Course Timetabling System

This Section is dedicated for analyzing the Military Technical College hybrid course timetabling system as a real case study. The timetabling problem at the MTC is a mix between school and university timetables. The constraints are very hard. The lectures are only scheduled during daytime. Forty teaching hours are available during the week. The number of teaching hours for any student is forty hours weekly. No vacant periods could happen. The problem is a very condense one.



Military Technical College
DTP (Planning Branch)

Ac. Year: 1997/98
Term: winter

Teaching Plan
4th. Year Electrical

Code	Subject Name	Type	Classes										
			HA	GU	CP	WW	EW	OE	AB	AG	AS	AA	EP
EE16	ELECTRONIC CIRCUITS	M-W	22	22	22	22	22	22	22	22	22	22	
EE23	ELECTRONIC MEASURMENTS	M-W-P	22	22	22	22	22	22	22	22	22	22	
EE36	MICROWAVE TECHNIQUE	M-W	41	41		41	41		41	41			
ES12	TELEPHONY	M-W-O				43							
ES37	NAVIGATION Th. & SYSTEMS	M-W-O							52	52			
ES41	SIGNAL THEORY	M-W-O					43						
ES52	LOGIC CIRCIUTS	A-W	22	22		22	22	22	22	22	22	22	22
ES54	RADAR THEORY II	M-WO	43										
ES62	GUIDANCE THEORY	M-W-O		43									
ES66	PULSE TECHNIQUE	M-W-O	41	41	41	41	41	41	41	41	41	41	
ES79	ASSEMBLY & Sys. PROGRAMMING	M-W			42								
ES729	COMPUTER Sys. ARCHITECTURE	M-W-O			42								
ES82	OPTICS	M-W-O						43					
ES817	AUTOMATIC CONTROL	A-W	41	41	41			41			41	41	41
ES818	AUTOMATIC CONTROL	A-W				41	41		41	41			
...	...												
...	...												
...	...												
ML12	MILITARY SUBJECTS	A-W-P	30	30	30	30	30	30	30	30	30	30	30
No. of Subjects			10	10	10	10	10	10	10	10	11	11	11
Total Lectures			27	27	28	27	27	26	26	28	25	27	24

Fig-2 *Teaching Plan4th. Year Electrical*

The Military Technical College (MTC) is one of the Institutions of the Arab Republic of Egypt (ARE) Armed Forces, responsible for the graduation of military engineers. It is organized as a set of academic branches. Each branch consists of several departments, which carry out the academic and instructional work. During daytime, students must attend four periods of lectures (two hours each,) and perform exercises in different college departments. No lectures or classes can be scheduled in the afternoon. So, all teaching activities must be carried out during the daytime. This constraint makes the problem of timetabling a very hard one. Branches and departments group the curricula for the offered subjects. Every curriculum is specified by the subject code, name and Lecture/Exercise weekly hours. For example, the course "ES709" *Assembly & Systems Programming* 4/2, means that, this subject or course "09" is given by the department of *Computer & Operations Research* "7" of the branch of *Applied Electrical Engineering* "ES" in "4" hours weekly as lectures and "2" hours weekly as exercises.

All courses are grouped in the teaching plans. The teaching plan is a table that lists the codes, weekly hours (WH) and total number of hours (TH) for every subject studied each semester. Fig-2 shows an example of one of the teaching plans.

Each staff member has his different loading hours per week. The loading hours ranges from 8 to 20 hours weekly. The administration decides the max no of loading hours to each staff member, this is due to the subjects he taught, his scientific degree and his location in the college.

4.2 The Course Scheduling Process

In any large university or institution there will be many difficult management and administrative problems of which timetabling is only one. Within the timetabling process, independent automated systems may prove useful to the administrators in many different situations. The timetabling process itself may be defined as the sequence of actions taken to produce a final timetable from students and courses data (see Section 2). This, of course, includes not only the actual scheduling stage where courses are assigned to timeslots but also consolations, checking and correcting data and document production.

In course timetabling, the typical sequence of events is roughly as follows. Departments decide which courses to be given. After several weeks when options are finalized, data is collated together. Room assignments may be separated or done at the same time; the scale of the room assignment problem is likely to be less than the course scheduling one. The draft timetable is then circulated to departments/staff who may make comments that are then taken into account. At this stage, the timetable is then presented to the instructors and students, who have a fixed deadline by which time they must report any problems (e.g. clashes that have not been previously noticed). Any problems are then dealt with (possibly by reconstructing part of the timetable.) and final timetable is produced. This is distributed to all students, staff and other personnel involved in the teaching process.

It is interesting to consider at this point "Just what is an appropriate use of computer processing in the management of a modern school or university timetable?" There is no doubt that each sub-process may be automated, but would it be an effective and appropriate use of the computer? This will obviously vary according to the size and requirements of the particular institution but many have reported good results. Unfortunately, the problems that cause systems to fail are rarely reported.

Although, from a computational point of view, the part of the timetable production forms the hardest part of the process, the other stages also provide their own problems, both administratively and logistically, and therefore should not be overlooked.

4.3 Description of the Timetabling Process at MTC

The Military Technical College is a specialized engineering college. It has very special specializations in different engineering branches. The number of these specializations is a much more than other engineering's faculty. This makes a numerous number of courses offered at the college.

The quality of teaching at Military Technical College is of a great importance. To achieve this quality, number of students per class can not exceed twelve students. Because all of the above, the timetabling process at Military Technical College is a very hard problem. The process goes in a sequence of phases as shown in Fig-3.

The teaching plan for a department is a list of all courses offered by this department in a given term. The extract of courses in a typical department is shown in Fig-4.

There are courses, such as *"Pulse Technique"*, *"Logic Circuits"*... are to be giving to several specializations. In such case the students may be divided into groups. The number of groups for each course based on number of students to study that course and the maximum number of students allowed setting together in an auditorium to listen to a lecture. After a department receives the extract of the teaching plan and number of groups for each course, the total number of hours for a course is decided.

For example, The course *"Pulse Technique" (4/1)*, as shown in Fig-4, is to be given to all Fourth Year Electrical. And the maximum allowed number of students in a group equals 50, which means that we have two groups in Fourth Year Electrical. Then the total load hours for that course will be 8 lecturing hours and 9 exercises hours. Each department board, should sit together to decide who will teach each course.

The teaching load form actually can be seen as a two dimensional array. The rows represent subjects, while the columns represent staff members. The instructors make their decision based on their interests, qualifications, availability and all other factors, the lecturing and exercising hours for some courses may be given by more than one instructor. After the teaching load form is being filled, personnel in charge of timetable preparation should revise it. The purpose of revision is to make sure that teaching hours of all courses offered in the given term has been assigned to instructors, and to make sure that the load is balanced between all staff members in the college. Also the assigned hours conforms to regulations informed by the teaching board of the college. Any problems or any thing that does not conform to rules should be discussed and solved before the next phase starts.

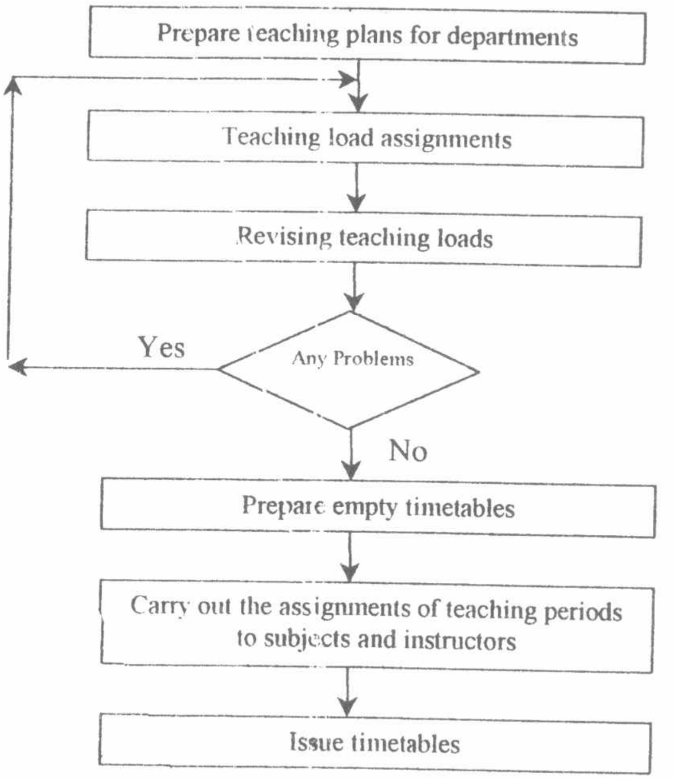
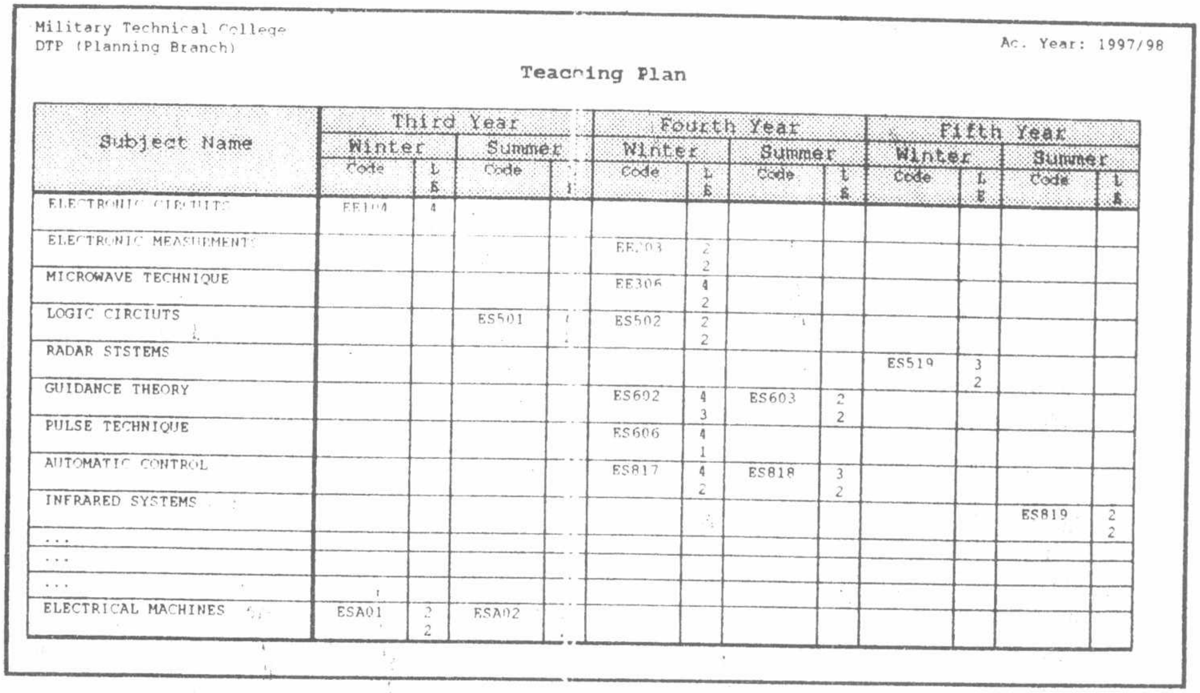

Prepare teaching plans for departments
Teaching load assignments
Revising teaching loads
Any Problems
Yes
No
Prepare empty timetables
Carry out the assignments of teaching periods to subjects and instructors
Issue timetables

Fig-3 *Timetabling Process* Phases

The empty timetable forms are prepared. The form can be seen as a matrix. The rows are labeled by the weekdays divided into lectures. Each day is divided into four periods. Each period is two teaching hours. The period splits into two parts, the upper stands for odd weeks and the lower stands for even weeks. If a course is to be given as three lecturing hours for certain specialization. Then it should occupy a complete period and a half in either odd or even weeks. Also if there is a course to be given for one hour per week, then the class will meet every other week, either in the odd numbered weeks or even numbered weeks. The columns in the form are labeled by all classes in the college.



Military Technical College
DTP (Planning Branch)

Ac. Year: 1997/98

Teacning Plan

Subject Name	Third Year				Fourth Year				Fifth Year			
	Winter		Summer		Winter		Summer		Winter		Summer	
	Code	L &	Code	L &	Code	L &	Code	L &	Code	L &	Code	L &
ELECTRONIC CIRCUITS	EE104	4										
ELECTRONIC MEASURMENTS					EE203	2 2						
MICROWAVE TECHNIQUE					EE306	4 2						
LOGIC CIRCIUTS			ES501		ES502	2 2						
RADAR STSTEMS									ES519	3 2		
GUIDANCE THEORY					ES602	4 3	ES603	2 2				
PULSE TECHNIQUE					ES606	4 1						
AUTOMATIC CONTROL					ES817	4 2	ES818	3 2				
INFRARED SYSTEMS											ES819	2 2
...												
...												
...												
ELECTRICAL MACHINES	ESA01	2 2	ESA02									

Fig-4 *Teaching Plan for Typical Department.*

Several timetable forms are printed out. A form for every group of related specializations is printed out. Typically, one form is printed for electrical specializations, another for mechanical and third one for other specializations then the most important phase in the manual process begins at this point. So at least two highly qualified staff members set together to start assigning the time slots to courses and instructors.

5. Modeling of the Considered Timetabling System

This section considers the modeling of the MTC timetabling system. The timetabling process encompasses every action that is required to create a final, complete and correct timetable from the initial data input. This includes receiving, validating and formatting data, sending out draft versions, making alterations and then producing the final timetable. In this section, we are going to discuss the data model of our timetabling process in the Military Technical College, the preprocesses or subprocesses which must be done before setting the timetable, the applied algorithm and the computer based resources which are used for implementing the timetabling process. In addition, how much time is available to produce the timetable and why might timetables change from year to year as well as between draft and final versions.

5.1 The Data Model

The data model can be introduced as an Entity Relationship Diagram (ERD) as shown in Fig-5. There are several main entities in any course timetabling system, the detailed description of these entities differs from one timetabling system to another depends on the nature of the colleges strategies. Whoever, a tentative list of these entities may include the Branch, the Department, the Class, the Course, the Instructor, the Room and the Timeslot.

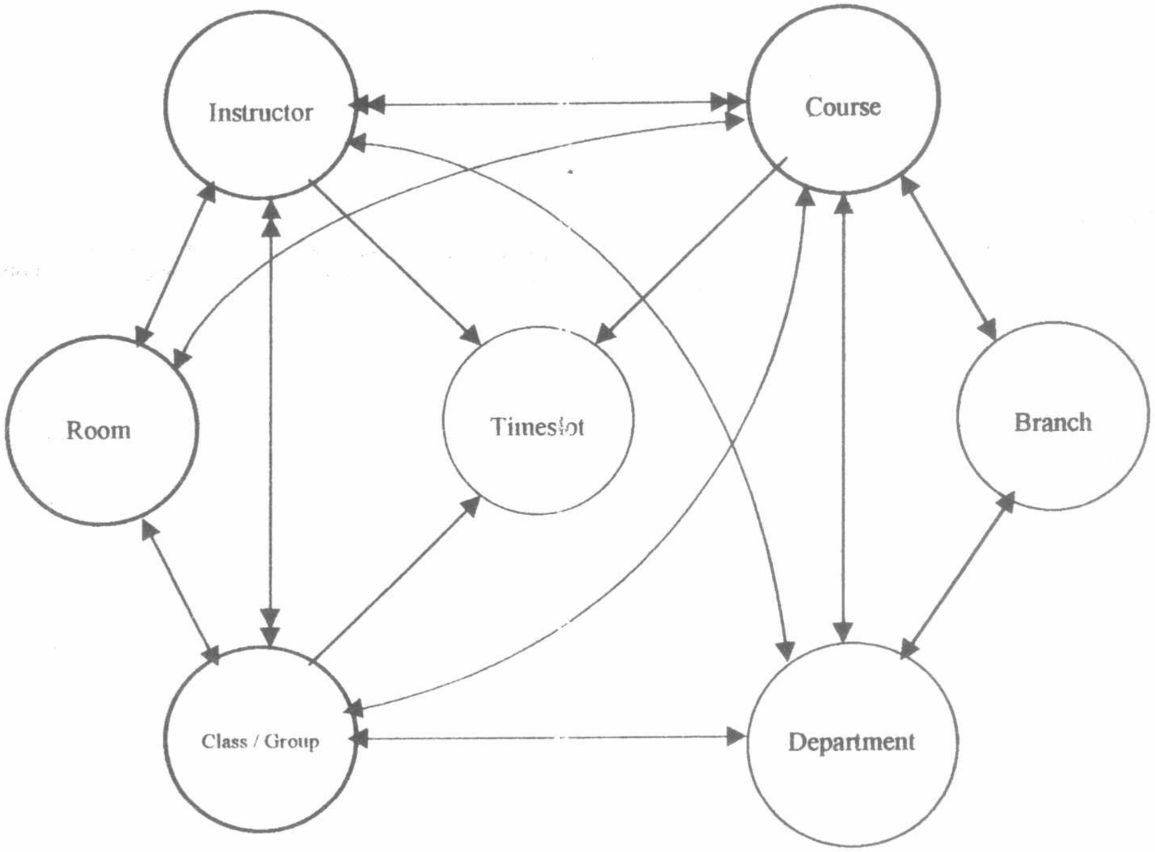


Fig-5: *Entity Relationship Diagram (ERD)*

As mentioned before in sections two and four, a valid timetable involves scheduling classes or groups, instructors, subjects and rooms into a fixed number of timeslots, in such a way that no instructor or class is used more than once per slot. In addition, in each timeslot a class is taught a course by an instructor. A particular combination of class, course, instructor and room is called a *tuple*. A tuple may be required more than once per week.

In any timetabling system, the core entity is the timeslot entity. The timeslot entity is not a physical entity. It results from the collection of four major entities, the class, the course, the instructor, and the room (bold border entities in Fig-5).

Based on the importance of these entities, the following subsections include detailed description of three of these entities.

5.2.1 The Class / Group

A class in the college should belong to one of the five academic years. classes in the first year are of general specialization. In the second year, a class should belong to one of the five main engineering branches in the college. A class in the third year to the fifth year should belong to one of the specialization departments of different engineering branches. Every class is given a unique code. Sometimes we have to collect some classes together forming a group. these classes have some common courses.

5.2.2 The Course

Each course in the college belongs to one of the eight Branches, and their departments. A specific department is responsible for teaching the planned lecture hours and planned exercise hours for the course by its own instructors, these planned hours is are determined by the teaching plans.

The course code is a combination of three codes. The first field is determined by the branch code (two letters). The department's code (two digits) specifies the second field, and the third field is a serial number for the course in the department (two digits). For example the *"Electronic Circuits"* course has the code *EE0706* which means that this is the course number 6 given by the *"Circuits & Systems"* department which is the department number 07 in the *"Electrical Engineering"* branch.

5.2.3 The Instructor

Like the course in the college, each department in the college branches has it is own instructors or staff. The department chief maneuvers with his staff in distributing the teaching load in a predefined ratio.

The instructor's load is decided according to many parameters like his scientific degree, military rank, his additional position and tasks in the college, and mainly his experts in department fields. The teaching load in some departments may be large number of hours according to extension of course materials or decreasing of the department staff member due to military job movements. In that case, the department starts to make contracts with external instructors from different universities to teach in the military technical collage and to share load with its staff members.

The instructor may not be available in either all weekdays or in all day hours. This may happen because of research activities or additional administrative tasks in the college.

5.4 The Process Analysis

This section introduces the process model of a general course timetabling system followed by a detailed description of the process model of the paper case study. We briefly described the three main entities of our problem, the class or group, the course and the instructor. Now we will discuss how these entities are tied together to perform the required tuples and then to assign a timeslot for each of these tuples.

A fully automated course timetabling process consists of four: subprocesses: the ***course planning*** subprocess which is most probably do not change from year to year. the ***class grouping*** subprocess which differs from year to year, the ***teaching load assignment*** subprocess which changes from one term to another in the same year and sometimes it changes within the same term, and the ***time slot assignment*** process. Fig-6 shows the process sequencing for the timetabling process including the three subprocesses. We will discuss how we handled each process in the automated system to make it just under user fingertips.

5.4.1 The Classes Grouping Subprocess

Based on teaching plans, the classes may be grouped together. This grouping should be done carefully and it takes a lot of time and effort if it is done manually. The maximum number of students in each group and the balance between number of students in all groups in the same specialization should be considered. It is always required to reduce number of groups to reduce the overall teaching load. A class may belong to more than one group in different subjects. This makes an additional constraint to the timetabling process. This makes the process of plotting adjacent classes in the empty timetable a difficult problem.

5.4.2 The Planning Subprocess

The planning process is the assigning of different courses to different specializations in five academic years with specification of weekly planned lecture and exercise hours. This process is done once ever, and sometimes the user needs to do little changes to these assignments, if he dose, he should do it before getting into the other processes.

5.4.3 The Loading Subprocess

The loading process is the assigning of teaching hours even lecture or exercise hours for specific course to a specific instructor or instructor. Each instructor belongs to a specific department, which is responsible to teach its courses with its instructors.

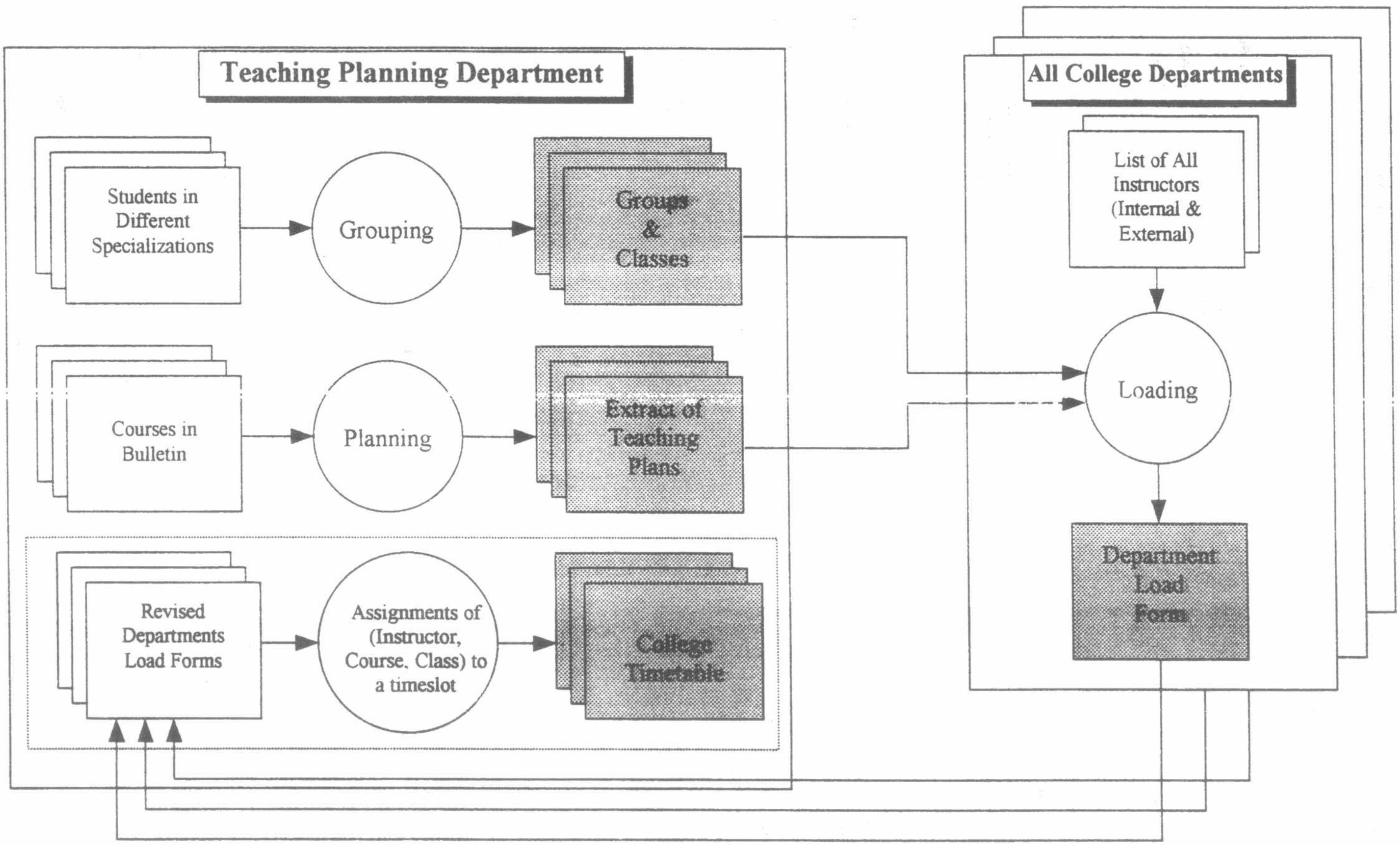

Teaching Planning Department
Students in Different Specializations
Grouping
Groups & Classes
Courses in Bulletin
Planning
Extract of Teaching Plans
Revised Departments Load Forms
Assignments of (Instructor, Course. Class) to a timeslot
College Timetable
All College Departments
List of All Instructors (Internal & External)
Loading
Department Load Form

Fig-6: *Course Timetabling Process Sequencing in the college*

5.4.4 The Time Slot Assignment Subprocess

This is the most difficult part of the problem. A set of heuristics is used while assigning timeslots in the manual system. This subprocess is a very time consuming one. In many automated systems, artificial intelligence techniques should be considered in solving such a problem.

6. Conclusion

The timetabling problem is a hard one. Different operations research techniques have been developed to tackle such a problem. Recently, artificial intelligence techniques and tools proved to be efficient in solving this problem.

This paper gives a study of the classes of the timetabling problem. The paper also gives a comprehensive study of a very special and complicated timetabling system at the Military Technical College, which gives a proposed model that, could be used for solving the problem. The model includes both data and process modeling.

A full implementation of an automated system will appear in a future work by the authors.

References

[1] **Abramson, D.** and **Abela, J.** (1992). *"A Parallel Genetic Algorithm For Solving The School Timetabling Problem"*. High Performance Computation Project, Division of Information Technology, C.S.I.R.O., Carlton, Australia. Published at the 15 Australian Computer Science Conference, Hobart, 1992.

[2] **Azvedo, Francisco** and **Pedro Barahona**. (1994). *"Timetabling In Constraint Logic Programming"*. In proceedings of World Congress on Expert Systems '94. **fa@fct.unl.pt**

[3] **Burke, E., David Elliman** and **Rupert Weare**. (1995). *"The Automated Timetabling Of University Exams Using A Hybrid Genetic Algorithms"*. Department of Computer Science, University of Nottingham, Nottingham, UK. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[4] **Burke, E., David Elliman, Peter Ford** and **Rupert Weare**. (1995). *"Examination Timetabling In British Universities - A Survey"*. 1st. International Conference on the Practice and Theory of Automated Timetabling (ICPTAT'95), Napier University Edinburgh. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[5] **Burke, E., David Elliman** and **Rupert Weare**. (1995). *"A Hybrid Genetic Algorithm For Highly Constrained Timetabling Problems"*. 6th. International Conference on Genetic Algorithms, Pittsburgh, USA, July 1995. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[6] **Burke, E., David Elliman** and **Rupert Weare**. (1995). *"Specialized Recombinative Operators For Timetabling Problems"*. Department of Computer Science, University of Nottingham, Nottingham, UK. **http** :// tawny. cs. nott. ac. uk/ ttg/index.html.

[7] **Burke, E., David Elliman** and **Rupert Weare**. (1994). *"A Genetic Algorithm For University Timetabling"*. In Proceeding of the AISB Workshop on Evolutionary Coputation. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[8] **Burke, E., David Elliman** and **Rupert Weare**. (1994). *"The Automation Of The Timetabling Process In Higher Education"*. Department of Computer Science. University of Nottingham, Nottingham, UK. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[9] **Burke, E., David Elliman** and **Rupert Weare**. (1994). *"A University Timetabling System Based On Graph Coloring And Constraint Manipulation"*. Journal of Research on Computing in Education. Vol. 27 issue 1, pp1-18. **email: ekb@uk.ac.nottingham.cs**

[10] **Burke, E., David Elliman** and **Rupert Weare**. (1994). *"A Genetic Algorithm Based University Timetabling System"*. In 2nd. East-West International Conference on Computer Technologies in Education. Crimea, Ukraine.

[11] **Burke, E., David Elliman** and **Rupert Weare** (1993). *"Automated Scheduling Of University Exams"*. Proceedings of I.E.E. Collequium on Resource Scheduling for Large Scale Planning Systems, Digest No. 1993/144. **http :// tawny. cs. nott. ac. uk/ ttg/index.html**.

[12] **Corne, D., Hsiao-Lan Fang** and **Chris Mellish**. (1993). *"Solving The Modular Exam Scheduling Problem With Genetic Algorithms"*. Proceedings of the 6th. International Conference in Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 370-373 .

[13] **Corne, D.** and **Peter Ross**. (1995). *"Peckish Initialization Strategies For Evolutionary Timetabling"*. Department of Artificial Intelligence, University of Edinburgh, Edinburgh, UK. In proceeding of the first international conference on the theory and practice of automated timetabling, Napier University, Edinburgh, 1995.

[14] **Corne, D., Peter Ross** and **Hsiao-Lan Fang**. (1994). *"Evolutionary Timetabling: Practice, Prospects And Work In Progress"*. In UK planning and scheduling SIG Workshop. email: **{dave| peter| hsiaolan} @ aisb.ed.ac.uk**

[15] **Frangouli, H., Vassilis Harmandas** and **Panagiotis Stamatopoulos**. (1995). *"UTSE: Construction Of Optimum Timetables For University Courses - A CLP Based Approach"*. Department of Informatics, Panepistimiopolis, University of Athens, Athens, Greece. **email: {bxpro| vassilis| takis} @di.uoa.ariadne-t.gr**

[16] **Ghosh, Sukumar** and **Mehmet Hakan Karaata**. (1991). *"A Self Stabilizing Algorithm For Coloring Planner Graphs"*. Department of Computer Science, University of Iowa, Iowa City, IA, USA. In the 29th. Allerton Conference on control. Communication & Computing In Oct. 1991.

[17] **Gunasena, U., Soundar R. Kumara** and **Allen L. Soyster**. (1989). *"A Knowledge Based System For Course Scheduling"*. Artificial Intelligence and expert systems Laboratory, Department of industrial and management Systems engineering, Pennsylvania State University. University Park, Pennsylvania. Applied Artificial Intelligence: 3: 463-482. 1989

[18] **Lajos, Gyuri** (1995). *"Complete University Modular Timetabling Using Constraint Logic Programming"*. In the 1st. International Conference on the Practice and theory of Automated Timetabling. Pages 364-375. **email: gyuri@scs.leeds.ac.uk**

[19] **Monfroglio, Angelo**. (1988). *"Timetabling Through A Deductive Database: A Case Study"*. Via Beldi 19, 28068 Romentino, Italy.

[20] **Ross, P., Dave Corne** and **Hsiao-Lan Fang**. (1994). *"Successful Lecture Timetabling With Evolutionary Algorithms"*. In Proceeding of the 11th. ECAI workshop. email: **{dave| peter| hsiaolan} @ aisb.ed.ac.uk**

[21] **Schaerf, Andrea**. (1995). *"A Survey Of Automated Timetabling"*. Computer Science/Department of Software Technology. Tech. Rep. CS-R9567. CWI. Amsterdam. The Netherlands.