Revista de Estudos Politécnicos
Polytechnical Studies Review
2008, Vol VI, nº 9

ISSN: 1645-9911

A System for Automatic Construction of Exam Timetable Using Genetic Algorithms

José Joaquim Moreira[1]
jmoreira@mail.islagaia.pt

(recebido em 14 de Abril de 2008; aceite em 5 de Maio de 2008)

Resumo. A complexidade do problema dos calendários de exames justifica-se pela dimensão do escalonamento de exames e pelo elevado número de restrições e critérios de atribuição. Este artigo apresenta um método de solução para o problema da construção automática dos calendários de exames. Entre vários modelos matemáticos de representação, a opção final foi por um modelo matricial, que é justificado pelas vantagens que este modelo apresenta quando utilizado no algoritmo de solução. O método de solução consiste numa meta-heurística que inclui um algoritmo genético. O modelo é direcionado para a construção de calendários de exames em instituições de ensino superior. Os resultados alcançados em reais e complexos cenários são satisfatórios; os calendários de exames cumprem as regulamentações impostas. Concluímos que, quando o algoritmo não determina uma solução, sem penalização, é porque essa solução não existe.

Palavras-chave: Escalonamento, Problemas de horários, Calendários de exames, Algoritmos genéticos.

Abstract. The complexity the of exam timetables problem is justified by the scheduling size of the examinations and the high number of constraints and criteria for allocation. This paper presents a solution method to the problem of automatic construction timetables for the exams. Among several mathematical models of representation, the final option was for a model matrix, which is justified by the benefits that this model presents when used in the algorithm solution. The method of solution is a meta-heuristics that includes a genetic algorithm. The model is directed to the construction of exam timetables in institutions of higher education. The results achieved in real and complex

[1] ISLA - Rua de Cabo Borges, 55, 4430-032 V. N. de Gaia, Portugal.

scenarios are satisfactory; the exam timetabling meets the imposed regulations. We conclude that when the algorithm does not determine a solution with no penalty, is because that solution does not exist.

Keywords: Scheduling, Timetabling problems, Exam timetabling, Genetic algorithms.

1. Introduction

Every school year, each exams period the different departments of an institution of education is facing the difficult task of drawing up timetables for examinations.

The difficulty is due to the great complexity of the construction of timetables for exams, due the scheduling size of the examinations and the high number of constraints and criteria of allocation, usually circumvented with the use of little strict heuristics, based on solutions from previous years.

The objective of this work is the examination schedules. The main purpose is to demonstrate the possibility of building them, automatically, by using computers.

The term scheduling applies to a kind of problem that, according Wren (Wren, 1995) distribute objects, subject to certain constraints, in a pattern of time or space, so that the costs of these are little. Objects may be people, vehicles, machines, exams, etc.., constraints are the rules that govern the process of scheduling, and some are inviolable while others take the form of principles that must be obeyed.

The problem of production of a factory described by Thompson (Thompson, 1995), the problem of traveling salesman approached by Wren (Wren, 1995) and the problem of school timetabling, with a solution proposed by Queirós (Queirós, 1995), for example, can be seen in perspective problems of sequential scheduling.

This subject has received special attention of the scientific community in the last five decades. This great interest, caused in 1995, the creation of series of conferences PATAT (Practice and Theory of Automated Timetabling) with new editions every two years (PATAT, 1995) and the establishment of EURO (Association of European Operational Research Societies) WATT (Working Group on Automated Timetabling). In 2002 emerged with the support of PATAT, the International Competition of Timetabling (ITC, 2002).

In this work, the genetic algorithm is the method of solution. Designed by John Holland (Holland, 1975) at the end of the fifties, uses a structure similar to that set out by Charles Darwin in "The Origin of Species'. It is based on two main operators: selection and reproduction, activated in the presence of a number of solutions, called population.

The formal model expresses using matrix representations. The application of genetic algorithm to matrix model creates exam timetables that meet the imposed regulations.

1.1 Related Works

Several works have approached the timetabling problem. Oliveira (Oliveira and Reis, 2000) presents a language for representation of the timetabling problem, the UniLang. UniLang intends to be a standard suitable as input language for any timetabling system. It enables a clear and natural representation of data, constraints, quality measures and solutions for different timetabling (as well as related) problems, such as school timetabling, university timetabling and examination scheduling.

Gröbner (Gröbner, 1997) presents an approach to generalize all the timetabling problems, describing the basic structure of this problem. Gröbner proposes a generic language that can be used to describe timetabling problems and its constraints.

Chan (Chan, 1997) discusses the implementation of two genetic algorithms used to solve class-teacher timetabling problem for small schools.

Fang (Fang, 1994), in his doctoral thesis, investigates the use of genetic algorithms to solve a group of timetabling problems. He presents a framework for the utilization of genetic algorithms in solving of timetabling problems in the context of learning institutions. This framework has the following important points, which give you considerable flexibility: a declaration of the specific constraints of the problem and use of a function for evaluation of the solutions, advising the use of a genetic algorithm, since it is independent of the problem, for its resolution.

Fernandes (Fernandes, 2002) classified the constraints of class-teacher timetabling problem in constraints strong and weak. Violations to strong constraints (such as schedule a teacher in two classes at the same time) result in a invalid timetable. Violations to weak constraints result in valid timetable, but affect the quality of the solution (for example, the preference of teachers for certain hours). The proposed algorithm, evolutionary, has been tested in a university comprising 109 teachers, 37 rooms, 1131 a time interval of one hour each and 472 classes. The algorithm proposed in resolving the scheduling without violating the strong constraints in 30% of executions.

Eley (Elay, 2006) in PATAT'06 presents a solution to the exam timetable problem, formulating it as a problem of combinatorial optimization, using algorithms Ant, to solve.

Analised the results obtained by the various works published, we can say that the automatic generation of schedules is capable of achieving. Some works show that when compared with the schedules manuals in institutions of learning real, the times obtained by the algorithms for solving the class-teacher timetabling problem are of better quality, since, uses some function of evaluation.

1.2 Organization of Paper

The concepts introduced in the Introduction are consolidated in the two chapters that follow. Thus, in Chapter 2, present the objectives of the exam timetables problem. Chapter 3 is devoted to the presentation of the method of solution used in this paper, describing the main concepts of Genetic Algorithms. Chapter 4, describes the activity of building exam timetables, it presents the model proposed, through its formalization, the model subjected to a simulated test with a simple problem, only for the purpose of demonstration and validation of the model and discuss the results, including results of larger problems and real. In Chapter 5, are pointed out the main conclusions and directions of future work.

2. Exam Timetables

The resolution of the exam timetables problem can be claimed by different areas, such as the School Administration, Artificial Intelligence, Mathematics or Operational Research. Probably, we must appeal on the techniques of simulation imported from fields as diverse as physics or biology, to solve the problem.
The purpose of the exam timetable is scheduler exams, according to pre-defined periods of time; minimizing losses teaching for the students, such as realize examinations on the same day or on consecutive days. But here, it considers each student individually, since the choice may depend only of the route of each school students.

The importance of the constraints, the quantity and quality, stems directly from the attempt to organize the problem. In this sense, we go classify, previously the constraints. Classified as constraints of the first order, or rigid, those are not being met, and it makes the scheduling illegal, calling themselve "impossible solutions". Other constraints, which should obey, and which, if not met, do not make illegal the scheduling, considered being of second order constraints, or flexible. So, we called the "impossible solutions" the scheduling, that check the constraints of the first order, Regardless of check, or not, the constraints of second order.

This division represents two moments in the resolution of the exam timetables problem. The first, consisting in the search for possible solutions, in the development of heuristics to ensure that the scheduling chosen corresponds to a possible solution. The second one consists on finding the best solution. The first runs in the space of all scheduling - which includes possible and impossible solutions; the second follows, just in the space of possible solutions.

3. Genetic Algorithms

The genetic algorithms distinguish themselves in the field of methods of optimization and search for the assimilation of the Darwinian paradigm of the evolution of species.

The genetic algorithms are processes of convergence (Queirós, 1995). Its structure is governed by import laws of the theory of evolution of species and concreteness in two fundamental concepts: selection and reproduction. The confrontation between genetic algorithms and the real problems is promoted by the need for optimization. It follows a space of enormous dimensions, in which each point represents a potential solution to the problem. In this maze of solutions, only a few, if not only one, fully satisfy the list of constraints that give shape to the problem.

The problems of optimization, usually associated with the satisfaction of constraints, define a universe of solutions, leaving the genetic algorithm to determine the overall solution, or a solution acceptable as a limitation on the time of action of the algorithm.

The genetic algorithms are search algorithms based on mechanisms of natural selection and genetics. Usually used to solve optimization problems, where the space of search is great and conventional methods is inefficient.

3.1 Characteristics

The terminology they are associated to translate the import of essential concepts of genetics and guesses the importance attributed to the interaction of these concepts. The concept of population, like number of individuals of the same species, is extended to artificial species. Individuals are normally represented by sequences of numbers: the genotype. The numbers, or rather, a collection of numbers, is the genetic heritage of the individual, determining their characteristics, that is, its phenotype. The genetic algorithms differ from traditional methods of research and optimisation, mainly in four aspects:

1. Work with a codification of the set of parameters and not with their own parameters;
2. Work with a population and not with a single point;
3. Uses information from or gain cost and not derived or other auxiliary knowledge;
4. Uses rules of transition probability and not deterministic.

The solutions interact, mix up and produce offspring (children) hoping that retaining the characteristics "good" of their ascending (parents), which may be seen as a local search, but widespread. Not only is the neighbourhood of a simple solution exploited, but also the neighbourhood of a whole population.

The members of the population are called individuals or chromosomes. As in natural evolution, the chromosomes are the base material (virtual, in this case) of heredity. It currently uses a function of evaluation that associates each individual, a real number that translates to adaptation.

Then, in a manner directly proportional to the value of their adaptation, are selected pairs of chromosomes that will cross themselves. Here, can be considered the selection with elitism, or ensure that the best solution is part of the new generation.

His crossing is the result of artificial selection, considering more adapted those that best meet the specific conditions of the problem. The crossing of the numerical sequences promotes the emergence of new sequences, formed from the first. With a probability established, after crossing, a mutation can happen, where a gene of chromosome changes.

These new individuals are the second generation of individuals and mark the end of cycle of the genetic algorithm. The number of cycles to perform depends on the context of the problem and the level of quality (partial or full satisfaction of the restrictions), which is intended for the solution.

A simple genetic algorithm describes the following cycle:

1st Generation of random n chromosomes that form the initial population;

2nd Assessment of each individual of the population;

3rd Verification of the termination criteria;

4th If verify termination criterion - cycle ending;

5th Selection of n/2 pairs of chromosomes for crossover;

6th Reproduction of chromosomes with recombination and mutation;

7th New population of chromosomes called new generation;

8th Go back to step 2.

The cycle described above is illustrated in Figure 1.

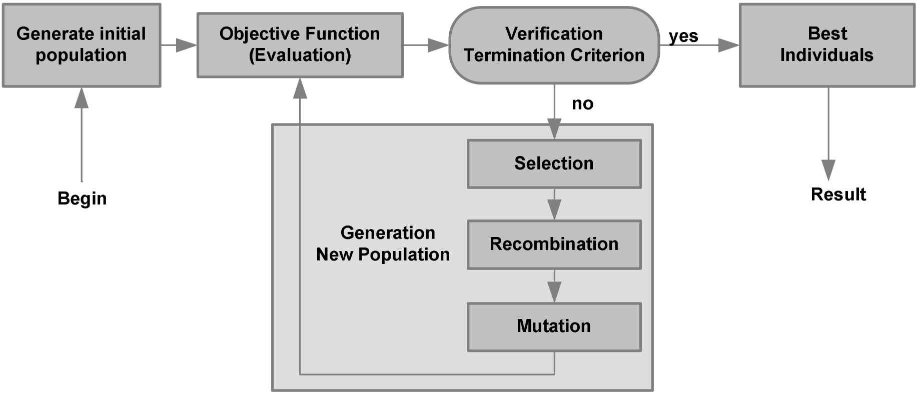

Generate initial population
Objective Function (Evaluation)
Verification Termination Criterion
yes
Best Individuals
no
Begin
Selection
Result
Generation New Population
Recombination
Mutation

Fig. 1. Basic structure of the genetic algorithm

4. Construction of Timetables for Exams

4.1 Model Proposed

The model we propose, matrix class, translates well the problem treated in this paper. It represents the allocation (or scheduling) of the examinations to the periods of time, supporting the limits that impose constraints conventional. The timetables are in the form of matrices of whole numbers, therefore easily manipulated by genetic algorithms.

4.1.1 Definitions

The construction of timetables for examinations requires the prior definition of some initial conditions. These conditions can be grouped into two broad areas: conditions of representation and conditions of constraints. The first is internal and coordinate the division of the periods of time and organization of resources; the second is external and limit the universe of scheduling.

As we presented the model proposed, it is subjected to a simulated test with a simple problem, only for the purpose of demonstration and validation of the model, the model is sufficiently broad to be confronted with real problems and more complex.

Conditions of representation

The scheduling of examinations assumes the prior organization of the days / periods of time that will be allocated exams. Admittedly, for example, that particular institution, for a certain period, defined two shifts - morning and afternoon - exams for day. Those periods being assigned, respectively, to the morning turn and part of the afternoon, one (10h) and two periods of time (14h and 17h) for the conduct of examinations. In Table 1 we have the distribution of periods of time each day.

Table 1. Distribution periods of time

Days	Turn	Period of time
1st	afternoon	14h e 17h
2nd	morning + afternoon	10h, 14h e 17h
3rd	morning + afternoon	10h, 14h e 17h
4th	morning	10h
5th	morning + afternoon	10h e 14h

This definition by the institution, result 2 x 3 + 2 x 2 + 1 x 1 (11) periods of time, where 1st period corresponds to the first day available for examination at 14h and 11th period corresponds to the last day available, at 14h.

For each of the subjects (13) that will be subjected to exams we have the subscription from each student, indicating the code of subjects and of the number of student enrolled.

Conditions of constraints

We will have to divide the conditions of constraints into two classes: first-order constraints and second order constraints.

Constraints of the first order:

1. A student may not have more than one examination on the same day;
2. Maximum number of examinations (classrooms available);
3. Preference of teachers (pre-marked examinations).

Constraints of second order:

1. A student should not have exams on consecutive days;
2. Examinations of a student evenly spread.

4.1.2 Representation Model Exam Timetables

Definition

H - set of all the periods of time that can occur examinations.

$H = \{h_1, h_2, ... h_m\}$, Where m corresponds to the maximum number of periods of time. In the previous example would: $H = \{1,2,3,...9,10,11\}$

Definition

D - set of all subjects, in a given season, will be under examination.

$D = \{d_1, d_2, ... d_k\}$, Where *k* is the maximum number of subjects, in a given season will be under examination. In the previous example would: *D* = {1101, 1102, 1103, 1104, 1105, 1106, 2201, 2202, 2203, 2204, 2205, 2206, 2207}

Model

A matrix M with 1 line and k columns that represent, in order, the subjects (examinations), of the D set, which will be scheduling. Each column contains a value withdrawn from of the H set, indicating the time at which the exam was allocated.

4.2 Application of the Model

Each subject, is given a serial number, according to the subscription of students in examinations. Thus, each matrix with 1 line and 13 columns (number of subjects) of elements, and for each column is permutations of H set, is a solution to the problem of timetables for examinations.

Although the problem presented be extremely simple, the space of candidate solutions to global solution comprises 34 522 712 143 931 different points (arrangements with repetition of eleven elements taken thirteen to thirteen). The growth of the variables in real problems increases the number of potential solutions for values even more enormous, being almost impossible a systematic evaluation to all solutions. Now, we present two solutions of exam timetable, obtained at random – table 2.

Table 2. Tow solution

$$\begin{matrix} \mathrm{M1} = [\\ \mathrm{M2} = [\end{matrix} \underbrace{\begin{matrix} 3 & 10 & 9 & 1 & 9 & 3 & 9 & 10 & 2 & 9 & 1 & 9 & 7 \\ 7 & 2 & 1 & 4 & 6 & 9 & 6 & 10 & 7 & 9 & 9 & 7 & 10 \end{matrix}}_{[\; d1 \;\; d2 \;\; d3 \;\; d4 \;\; d5 \;\; d6 \;\; d7 \;\; d8 \;\; d9 \;\; d10 \;\; d11 \;\; d12 \;\; d13 \;]} \begin{matrix}] \\ ] \end{matrix}$$

In which each d_n is replaced by subject (examination), whose serial number corresponds to the index of d. To assign the examinations to the respective periods of time we used the following examinations mask for the allocation of the periods of time (H), table 3.

Table 3. Mask of examinations for periods of time (H)

	1st day	2nd day	3rd day	4th day	5th day
10h		H3(3)	H6(6)	H9(9)	H10(10)
14h	H1(1)	H4(4)	H7(7)		H11(11)
17h	H2(2)	H5(5)	H8(8)		

Thus, the first solution presented (M1), result in the following schedule of exams - Table 4. Example: subject d1 (1001) allocated in position 3 (H3), etc.

Table 4. Calendar of examination for the solution M1

	1st day	2nd day	3rd day	4th day	5th day
10h		1101 1106		1103 1105 2201 2204 2206	1102 2203
14h	1104 2205		2207		
17h	2203				

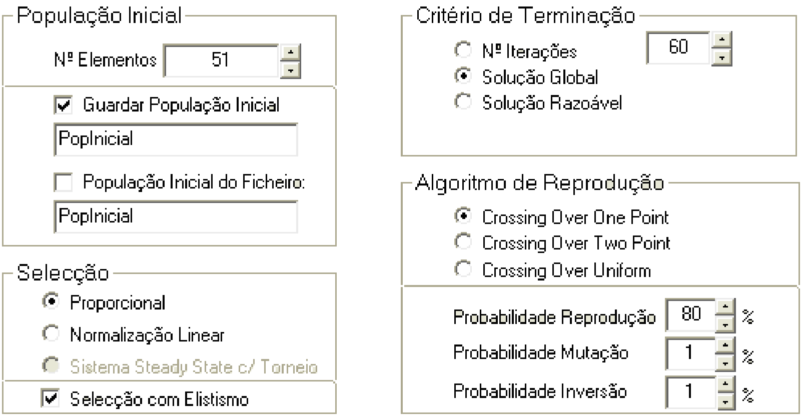
The next step of the algorithm is to evaluate each of the solutions (calendar) through a function of evaluation (1).

$$f(c)= P_1R_1 + P_2R_2 + P_3R_3 + P_4R_4 + P_5R_5 \tag{1}$$

Where P1, P2, P3, P4 and P5 are the value of the penalty for each constraint. R1, R2, R3, R4 and R5 represent the number of times the calendar *c* violates the restrictions 1,2,3,4 and 5, respectively.

Each solution, now is associated a numerical value that reflects their adaptation to the environment, or the conditions of constraints. The next stage that follows by the genetic algorithm, consist in the selection of n individuals, possibly with repetition, can also suffer mutation.

Before starting the computer generation of the schedule of examinations, it is necessary to customize the genetic algorithm. In the next figure we have the elements of customization of computational application – prototype.



População Inicial

Nº Elementos 51

☑ Guardar População Inicial

PopInicial

□ População Inicial do Ficheiro:

PopInicial

Selecção

● Proporcional

○ Normalização Linear

○ Sistema Steady State c/ Torneio

☑ Selecção com Elistismo

Critério de Terminação

○ Nº Iterações 60

● Solução Global

○ Solução Razoável

Algoritmo de Reprodução

● Crossing Over One Point

○ Crossing Over Two Point

○ Crossing Over Uniform

Probabilidade Reprodução 80 %

Probabilidade Mutação 1 %

Probabilidade Inversão 1 %

Fig. 2. Customization of the genetic algorithm – Prototype

4.3 Evaluation

Test scenario

An institution with 4 courses and a total of 77 subjects. 250 students are enrolled in examinations. Many students are enrolled in tests of previous years (delayed subjects).

There were created two instances of the problem for testing. Then each instance was submitted to the prototype.

- Instance 1 – 45 days with 3 periods of time

□ Instance 2 – 32 days with 3 periods of time

The computer of test had a 1.0 GHz Pentium III processor with 384 MB of RAM memory. The customization basis of the genetic algorithm was that is represented in figure 2. In each test, only the number of elements (solutions) of the initial population changed.

4.3.1 Results

In the various executions of the algorithm, we observed that the evolution of penalties from iteration to iteration (cycle of the algorithm) had a downward behavior - figure 3.

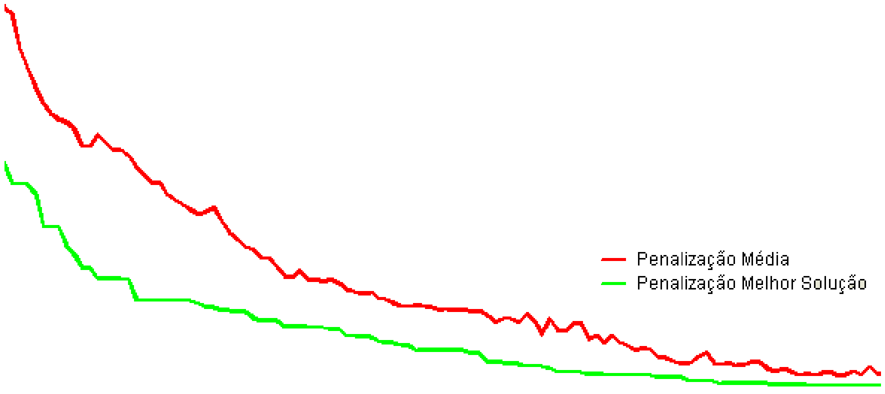


Fig. 3. Penalties evolution

In tables 5 and 6, have the results for the instances 1 and 2, respectively.

Table 5. Results of first instance

	Elements of the initial population	Penalty of 1st iteration (cycle)	Iteration (cycle) of the solution with penalty zero	Time
Test 1	51	12080	137	50 min
Test 2	101	8020	118	120 min

Table 6. Results of second instance

	Elements of the initial population	Penalty of 1st iteration (cycle)	Iteration (cycle) of the solution with penalty zero	Time
Test 1	51	14180	233	240 min
Test 2	101	10760	180	180 min

Increased amounts of elements (solutions) of the initial population, the one hand, it is more demanding for the computer, but, on the other hand, the overall solution is found with fewer iterations of the algorithm.

5. Conclusions and Future Work

The problem studied in this work, together with school timetabling, belongs to the class of more complex problems of combinatorial optimization with satisfaction of constraints. Its importance is well measured at each time of examinations, in each school year.

Our proposal focused on the preparation automatic exam timetables using computers. We define the problem and define the method of solution through genetic algorithms.

The foundations for the construction of an automatic generator has completed to the definition of an automatic model that represents the problem and the structure of the genetic algorithm. The model chosen, of nature matrix presents, in addition, other advantages: it is based on rigorous mathematical definitions, adjusting to an efficient analysis of the quality of the calendars, each matrix represents a solution (possible or impossible) and its elements belong to the set of integer numbers.

Under these conditions it has developed a prototype of automatic generation of schedules of examinations. All functions of the genetic algorithm were coded in the Visual Basic language.

The results achieved in all tests performed with real scenarios, in general, are satisfactory. The schedules of examinations meet the regulations imposed. When the algorithm does not determine a solution with zero penalty, can be explained by two reasons: this solution does not exist, that is, the overall solution admits some penalty, and / or the occupation, with examinations of all periods of time, that is, the inability to move an examination without changing the allocation of another examination.

In addition to the natural advantages in automating the process of the construction of timetables for examinations, it should be noted, also, the facilities at the editing of data that includes automation.

This work raises some clues for subsequent searches that can be topped. Thus, in the short term, the contemplation of the scheduling of examinations take into consideration the type of classrooms and also specify the number of vigilant required for each exam. In the long term, reconsidering adaptive techniques and confronting the results of three types of algorithms: genetic algorithms, tabu search algorithms and simulated annealing algorithms.

References

Chan, H. W (1997). *School Timetabling Using Genetic Search*. 2th International Conference on the Practice and Theory of Automated Timetabling, PATAT'97.

Eley, M. (2006). *Ant Algorithms for the Exam Timetabling Problem*. 6th International Conference on the Practice and Theory of Automated Timetabling, PATAT'06.

Fang, H. L. (1994). *Genetic Algorithms in Tametabling Problems*. PhD Thesis, University of Edinburgh.

Fernandes, C. (2002): *Infected Genes Evolutionary Algorithm for School Timetabling*. WSES International Conference.

Gröbner, M., Wilke P (2002). *A General View on Timetabling Problems. 4th International Conference on the Practice and Theory of Automated Timetabling* - PATAT'2002.

Holland, John (1975). *Scheduling, Adaptation in Natural and Artificial Systems.* The University of Michigan Press.

ITC (2002). *International timetabling competition*:
http://www.cs.qub.ac.uk/itc2007/index_files/overview.htm

Oliveira, E., Reis L.P. (2000). *A Language for Specifying Complete Timetabling Problems*. 3th International Conference on the Practice and Theory of Automated Timetabling PATAT'2000.

PATAT (1995). T*he International Series of Conferences on the Practice and Theory of Automated Timetabling (PATAT)* - http://www.asap.cs.nott.ac.uk/patat/patat-index.shtml

Queirós, F. H. (1995). *Construção automática de Horários de Aulas*. Tese de Mestrado, Universidade Portucalense.

Thompson et al, Jonathan e Dowland, Kathryn (1995). *General Cooling Schedules for a Simulated Annealing Based Timetabling System*. Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling.

Wren, Anthony (1995). *Scheduling, Timetabling and Rostering — a special relationship?* Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling, 474-495.