Managing the tabu list length using a fuzzy inference system: an application to exams timetabling

Tiago Cardal Pais * Paula Amaral †

30th October 2008

Abstract

In this paper we present an application of Tabu Search (TS) to the exams timetabling problem. One of the drawback of this meta-heuristic is related to the necessity of tuning some parameter (like tabu tenure) whose choice affects the performance of the algorithm. The importance of developing an automatic procedure is clear considering that most of the users of timetabling software, like university and school staff, do not have the expertise to conduct such tuning. The goal of this paper is to present a method to automatically manage the memory in the TS using a Decision Expert System. More precisely a Fuzzy Inference Rule Based System (FIRBS) was implemented to handle the tabu tenure based on two concepts "frequency" and "inactivity". These concepts are related to the number of times a move was attempt and the last time it was called. Computational results show that the implemented FIRBS handles well the tuning of the tabu status duration improving as well the performance of Tabu Search.

Key word: Exams timetabling; Tabu Search; Fuzzy Inference System.

1 Introduction

1.1 The examination timetabling problem

Problems related to timetabling are present in daily life. Solving timetabling problems is a crucial task and affects many institutions and services like hospital, transportation enterprizes, educational establishments, among many others. These problems have been an object of increasingly interest by the research community. Mainly in the field of Operations Research and Artificial Intelligence many interesting proposals have been presented, to solve timetabling problems in sports [19],[43], transportations (bus,railways,planes) [28], [7], [26], [33], schools [1], [15], [35], [27], [39] and universities [2], [4], [6], [8], [13], [11], [10], [16], [17], [18], [20], [21], [22], [23], [26], [30], [31], [32], [34], [36], [41], [42], [44], [45].

A general definition of timetabling was given by Burke, Kingston and Werra [5].

" A timetabling problem is a problem with four parameters, T a finite set of times, R a

*CA3 - UNINOVA, Quinta da Torre Campus FCT-UNL 2829 - 516 Caparica Portugal, tpp@uninova.pt
†Dep. Mathematics, Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal

finite ser of recourses, M, a finite set of meetings: and C, a finite set of constraints. The problem is to assign times and recourses to the meetings so as to satisfy the constraints as far as possible."

If in particular we consider exams as meetings then we are dealing with exams timetabling, which is the problem we are addressing in this paper. This is not a limitation regarding the major contribution of this paper since the ideas presented here can be extended to applications of Tabu Search to other timetabling problems. One of the reasons why exams timetabling problems have attracted a considerable interests over the last decades is justified by it importance and relevance in educational success. In the last years we have noticed a tendency towards the flexibility of curricula and the increase of the number of students enrolled in each course. These facts have introduced an additional source of difficulty in an already hard problem. The more transversal are the courses, the harder is to find a feasible examination timetabling. Spreading consecutive exams as much as possible allows for more time of preparation and usually students regard this feature as the most important, which if fully considered may increase the duration of the examinations calendar. On the other hand teachers in general tend to favor the shortage of the period for examinations to allow more time for preparing the next semester and do research. To balance conflicting interests is one of the difficulties to be faced when addressing this problem, but by far not the only one. The structural problem is to find a calendar without conflicts in the sense that students should not be schedule to take more than one exam at the same time. In addition there are other constraints that complicate the problem such as those regarding teachers, invigilators, rooms, precedences between exams, allowance of time for grading exams, and so on. This makes clear why exams timetabling together with classes timetabling are one of the most hard and time consuming tasks in scholar management.

1.2 Contributions of this paper

In this paper we propose to solve exams timetabling problems using Tabu Search in a completely automatic way. It is well known that in order to use Tabu Search it is required the tuning of some parameters in particular the duration of the tabu status. It is also well known that this task is cumbersome requiring in general the intervention of an expert. Computational implementations to solve real problems creates a demand for automatic solution methods. Tabu Search is a flexible, easy implementable and efficient method. To overcome the major drawback of TS in the perspective of automatization we proposed to use an intelligent system to perform the role of the expert in the task of tuning the parameters. To accomplish this goal we incorporated inside the TS a Fuzzy Inference Rule Based System (FIRBS). The design of this FIRSB took in consideration the characteristics of the problem but is independent of the particular instances.

1.3 Structure of the paper

The rest of the paper is structured in the following way. In Section 2 we give a literature review focusing only on examination timetabling. In Section 3 we present the mathematical formulation of the problem, a brief introduction on TS and a general overview of FIRBS comprising some basic definitions related to Fuzzy theory. In Section 4 we give the implementation details regarding the Tabu Search and the FIRBS. In Section 5 we present computational results focusing on the comparison with other methods to show the effectiveness of this method. Some conclusions are given in Section 6.

2 Literature Review

We can find in the literature a great number of excellent contributions to exams timetabling problems (Werra 1985; Carter 1986; Carter and Laporte 1996; Burke, Jackson et al. 1997; Schaerf 1999; Petrovic and Burke 2004). More recently it was published a very interesting technical report giving a general overview on the more recent approaches to this problem (Qu, Burke et al. 2006). To present the state of the art of some of the most successful contributions in the literature we adopted a classification proposed by Carter e Laporte (Carter and Laporte 1996) and later completed by Petrovic e Burke (Petrovic and Burke 2004) based on the methodology applied.

2.1 Clustering methods

Clustering methods for timetabling problems were, to our knowledge, first proposed by Desroches et al (Desroches, Laporte et al. 1978). These methods comprehend in general two stages. In a first stage exams are clustered and then the clusters are assign to real periods. In a second phase an improvement of the solution is attempted by interchanging exams between periods. Arani and Lotfi (Arani and Lotfi 1989)in their work proposed a multi-stage approach.

2.2 Sequential methods

The examination timetabling problem can be modeled as a node coloring problem in a graph. For this reason most of the heuristics for nodes coloring can be applied to this problem. These heuristics are often characterized by a construction phase where a feasible solution is obtained, where a key issue is the ordering strategy. The most common strategies are: Saturation degree (Brelaz 1979), Largest Degree (Broder 1964), Largest Weighted Degree (Carter, Laporte et al. 1996), Largest Enrolment" (Wood 1968) and Color Degree (Carter, Laporte et al. 1996). Asmuni et. al. (Asmuni, Burke et al. 2004) used a Fuzzy Inference System to order the exams.

2.3 Constraint based methods

Constraint based methods are competitive due to the flexibility of these approaches but accordingly to Brailsford et. al. (Brailsford, Potts et al. 1996) they are more successful if applied in combination with other methods like local search. Merlot et. al. (Merlot, Boland et al. 2003) used constraint programming to build an initial solution as a starting point for the application of Simulated Annealing and Hill Climbing.

2.4 Meta-heuristics

Due to the huge variety of approaches in this group we will consider its subdivision into more specific methodologies.

2.4.1 Tabu Search

Di Gaspero e Schaerf (Gaspero and Schaerf 2001) applied TS to the exams timetabling problem using a dynamic weight updated objective function. Later Di Gaspero (Gaspero 2002) has improved this approach using 3 different neighborhood named: Recolour, Shake and Kick. White and Xie (White and Xie 2001) presented OTTABU, a TS approach where the initial solution is generated using the method *Largest Enrolment* and both short memory and long memory strategies where used together with a memory relaxation for introducing diversification in the search.

2.4.2 Simulated annealing

Thompson and Dowsland (Thompson and Dowsland 1998) presented the application of SA to the exams timetabling problem. They proposed different cooling methods together with a set of neighborhoods (Simple and S-chains). In a previous paper the same authors (Thompson and Dowsland. 1996) showed that an adaptive cooling did in general produce a better result when compared with a geometric cooling strategy and that Kempe Chains (a particular case of the S-chains) performed better than the simple neighborhood.

2.4.3 Great Deluge

Great Deluge (GDA) was proposed by Dueck (Dueck 1993) as an alternative to SA, gaining in the reduced number of parameters and consequently avoiding the difficult task of tuning them. Burke and Newall (Burke and Newall 2003) conduct a comparison study between GDA and SA, showing the over performance of GDA. Burke et al (Burke, Bykov et al. 2004) implemented a SA and a GDA where the quality of the solution was measured according to the satisfaction of soft constraints. They also introduced two user defined parameters to control the execution time and the maximal quality of the solution. These parameters were used in a function that controlled the performance of the method.

2.4.4 Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) was first proposed by Mladenovi and Hansen (Mladenovi and Hansen 1997). The main new concept regards the existence of more than one neighborhood structure simultaneously, allowing to alternate between them in each run of the method. An application to the exam timetabling problem was proposed by the same authors (Hansen and Mladenovi 2001). More recently Burke et al (Burke, Eckersley et al. 2006) applied the VNS not as a local search but as an hyper-heuristic.

2.4.5 Genetic Methods

Erben (Erben 2001) developed a Grouping Genetic Algorithm for the node coloring problem and consequently with applications to the exams timetabling problem. The main novelty of this proposal is the definition of a chromosome as a group of genes, allowing for changes in the number of genes, contrarily to classical applications where this number is fixed.

2.5 Ant Colony

Costa e Hertz (Costa and Hertz 1997) developed a method based on Ant Colony, the ANTCOL, for the node coloring problem and it was suggested the application to the timetabling problem. Dowsland e Thompson (Dowsland and Thompson 2005) followed this suggestion applying ANTCOL to timetabling problems and enhanced the use and tuning of some parameters.

2.6 Memetic Search

Memetic Serach (MS) combines evolutionary algorithms with local search. Burke e Newall (Burke and Newall 1999) apply this idea in a decomposition strategy, consisting in dividing exams in two groups. One group was treated separately and fixed while the assignment of the remaining exams was done using MS.

2.6.1 Other meta-heuristics

There are some methodologies that can hardly be incorporated in the previous ones and that are worth mentioned due to their success in solving the examination timetabling problem. Among them we can mentioned the work of Caramia, Dell'Olmo e Italiano (Caramia, Dell'Olmo et al. 2001). These authors used a greedy method to assign the exams to the smallest possible number of periods using a technique named *Penalty Trader*. Abdullah et al (Abdullah, Ahmadi et al. 2006) developed an algorithm based on the Ahuja e Orlin neighborhood (Ahuja, Orlin et al. 2001). Even if computational heavy this method presents some of the best results for a collection of instances frequently used by many researchers in timetabling problems.

3 Preliminaries

3.1 Mathemathical formulation of this problem

The exams timetabling problem can be formulated as a combinatorial optimization problem, or an integer programming problem depending mainly in the constraints we want to introduce and the objective function. In order to be able to compare our approach with other methods proposed in the literature we adopt a common integer formulation [38] followed by many authors. Let,

$$\begin{aligned} K &= \text{number of courses,} && (1)\\ P &= \text{number of slots,} && (2)\\ c_{ij} &= \text{number of students enroled in course } i \text{ and } j \text{ for } i,j = 1,\ldots,K, && (3)\\ A_{ij} &= \begin{cases} 1 & \text{if } c_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i,j = 1,\ldots,K, && (4) \end{aligned}$$

and consider the following variables

$$\begin{aligned} x_i &= \text{slot to which the exam } i \text{ is assigned } i = 1,\ldots,K, && (5)\\ y_{ij} &= \begin{cases} 16 & \text{if } |x_i - x_j| = 1 \\ 8 & \text{if } |x_i - x_j| = 2 \\ 4 & \text{if } |x_i - x_j| = 3 \\ 2 & \text{if } |x_i - x_j| = 4 \\ 1 & \text{if } |x_i - x_j| = 5 \\ 0 & \text{otherwise .} \end{cases} \quad \text{for } i,j = 1,\ldots,K, && (6) \end{aligned}$$

The formulation is:

$$\begin{aligned} \min f &= \frac{\sum_{i=1}^{K}\sum_{j=1}^{K} c_{ij} y_{ij}}{2M} && (7)\\ \text{subject to} \quad & |x_i - x_j| \geq 1 - (1 - A_{ij})M_0 \text{ for } i,j = 1,\ldots,K, i \neq j && (8)\\ & 1 \leq x_i \leq P \text{ and integer } \text{ for } i,j = 1,\ldots,K, && (9) \end{aligned}$$

Where M_0 is an arbitrarily large number. The objective function (7) penalizes the proximity of exams with students in common, using as weights the number of students involved in both examinations and a factor that depends of the proximity of the periods, ranging from 16 to 0. The constrains (8) ensure that any two exams indexed by i and j with students in common are not assign to the same period. This problem is known to be NP-hard and global optimization procedures are in general unsuccessful for larger problems. We are aware of the importance of a globally optimal solutions, nevertheless the construction of good feasible solutions is plainly justified given the fact that the problem is hard, and sometimes the mathematical approach is not able to fully characterize all the aspects of the problem. In addition global optimization procedures are not abundant in the literature, while creative and successful heuristic approaches have been widely explored [5], [4], [8], [13], [45]. Tabu search is one of such methods that have been applied to solve these problems.

3.2 Tabu Search

Tabu search [24], [25] is a meta-heuristic that has successfully been applied to find good feasible solutions for hard optimization problems. In general it can be described as a neighborhood search method incorporating techniques for escaping local optima and avoid cycling. A fist level Tabu Search (TS) comprises the following concepts in each iteration:

- Current starting solution - Start search point.
- Search Neighborhood - Points that will be inspected from the current solution.
- Move - A basic operation in the definition of the neighborhood.
- Evaluation - A procedure to evaluate the points in the neighborhood.
- Tabu list - The tabu moves that are not allowed in the current iteration
- Aspiration Criteria - May revoke a tabu status.

A general, very basic, iteration of TS will consist in finding a set of points in the neighborhood of the current point. Evaluated these points and chose the one that has the best evaluation, as long as the move associated to this point is not tabu. If it is tabu we can apply an aspiration criteria or not. Next we add the move, or solution, or a related attribute that generated the best evaluated point to the tabu list. The number of iterations during which this move is kept in the tabu list may follow different strategies. Some of the most common are

- Keep the dimension of the tabu list fixed and limited.
- Assign a value for the tabu tenure equal for all moves.
- Assign different values of permanence in the tabu list depending on historical record.

After we proceed to the next iteration from the current point. There are many interesting additional refinements that can greatly increase the performance of TS. In this work we have chose to develop the most simple TS method to avoid masking the true effect of the FIRBS by introducing additional features.

3.3 Fuzzy sets, Linguistic Variables, Fuzzy Inference Rule Based System

Fuzzy systems apply $IF - THEN$ rules and fuzzy operators to develop some action, translated from the output of the system, given a certain input. Before understanding a Fuzzy Inference Rule Based System (FIRBS) it is necessary to introduce a few definitions namely of Fuzzy Sets and Linguistic Variables. Zadeh (Zadeh 1965) developed the theory on Fuzzy Sets to extend the classical crisp concept where an element either belongs or not to a certain set.

3.3.1 Fuzzy Sets

Definition 1. *(Zimmermann 1996): If $\mathcal{X}$ is a collection of objects designated by x then a fuzzy set $\widetilde{A}$ in X is defined by a set of pairs:*

$$\widetilde{A} = \{(x, \mu(x))|x \in X\}$$

where $\mu_{\widetilde{A}}(x)$ is the membership function of x em $\widetilde{A}$.

As an example we can consider the age of a person. Let X be the age domain and x the age of a certain person. Then the fuzzy set YOUNG may be defined by:

$$\widetilde{A} = \{(x, \mu(x)) | x \in X\}$$

where

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & \text{, if} \quad x \geq 65 \\ \frac{65-x}{30} & \text{, if} \quad 35 \leq x \leq 65 \\ 1 & \text{, if} \quad x \leq 35 \end{cases}$$

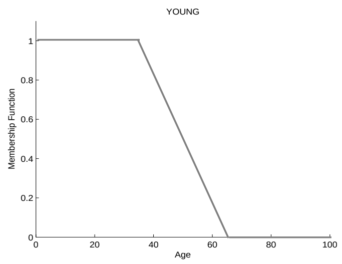

Figure 1: Membership function for fuzzy set YOUNG

Fuzzy sets are often represented by triangular, trapezoidal (triangular as a particular case) or gaussian membership functions. The membership function for YOUNG just presented is an example of a trapezoidal function. Generalizing, a trapezoidal function (Figure 2) is given by the following membership function:

$$\mu_{\widetilde{A}} : D \subset X \rightarrow [0, 1]$$

where

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & \text{, if} \quad x \geq a \\ \frac{x-a}{b-a} & \text{, if} \quad a < x \leq b \\ 1 & \text{, if} \quad b < x \leq c \\ \frac{d-x}{d-c} & \text{, if} \quad c < x \leq d \end{cases}$$

where $x \in D \subset \mathcal{X}$ and $\widetilde{A}$ is a fuzzy set in $\mathcal{X}$. The triangular function is a trapezoidal

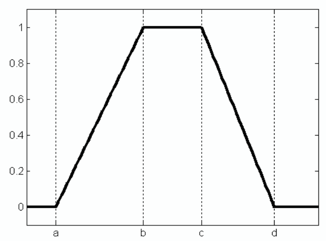

Figure 2: Trapezoidal membership function

function where $b = c$.

The gaussian function (Figure 3) is defined by $\mu_{\widetilde{A}} : D \subset \mathcal{X} \rightarrow [0,1]$ where

$$\mu_{\widetilde{A}}(x) = \exp^{\frac{x-c}{2\sigma}}$$

and where $x \in D \subset \mathcal{X}$, $\widetilde{A}$ is a fuzzy set in $\mathcal{X}$, c is the symmetry center and σ is the

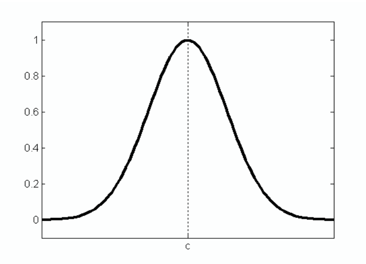

Figure 3: Gaussian membership function

function amplitude.

3.3.2 Linguistic Variable

The linguistic values or terms of a linguistic variables are concepts defined by words or expressions of a natural language.

Definition 2. *(Zadeh 1975): A linguistic variable is characterized by the quintuple* $(\mathcal{H}, T(\mathcal{H}), U, G, M)$ *, where* $\mathcal{H}$ *is the name of the variable,* $T(\mathcal{H})$ *is the set of terms or linguistic values of* $\mathcal{H}$*,* U *is the universe of the variable,* G *the semantic rule that generates the terms in* $T(\mathcal{H})$ *and* M *is the semantic rule associating to each term or linguist value its meaning trough the fuzzy set* $M(X)$ *(*$M(X)$ *is a fuzzy set on* U*).*

Let us consider the linguist variable TEMPERATURE as in Klir and Yuan (1995). We can have a pure numerical interpretation for this concept as depicted in case (b) in Figure 4, but we can represent it as a linguist variable (case (a)), characterized by the linguist values { Very Low, Low, Average, High, Very High}.

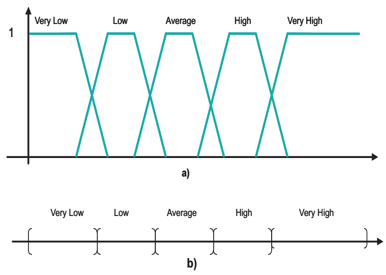

Figure 4: a) Temperature as Linguistic Variable - b) Numerical representation of Temperature

3.3.3 Fuzzy constraint

Definition 3. *(Zimmermann 1996): Let* A *be a fuzzy set in* U*, characterized by the membership function* $\mu_A(x)$*.* F *is a fuzzy constraint on the variable* X *if* F *works as an* *elastic constraint on the values that can be assign to X, in the sense that the values x that can be assign to X are $X = x : \mu_A(x)$.*

To exemplify we can consider the example "AGE" given above and the fuzzy constraint *John is YOUNG.* If for instance John is 25 years old then this age has a degree of compatibility with the fuzzy set YOUNG of 1.

3.3.4 Fuzzy Ruled Based Systems (FRBS)

Rule Based Systems (RBS) are a field of Expert Systems. A rule based system can generically be defined by a set or rules of the type

Rule i : IF X THEN Y.

and the goal is to emulate the behavior of human reasoning. In order to deal with imprecise concepts, which in some ways are closer to human reasoning, Zadeh introduced the theory of "Approximate reasoning":

"Informally, by approximate or, equivalently, fuzzy reasoning we mean the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises (Zadeh 1979)."

Approximate reasoning, based on Fuzzy Logic, was applied in the context of RBS given rise to the concept of Fuzzy Rule Based Systems (FRBS). Several models were proposed for implementing FRBS, Mandami (Mandani and Assilian 1975) is the most classical but later, a model known as Sugeno ou TSK (Takagi, Sugeno and Kang) [37], [40] gain popularity due to its simplicity regarding computational implementation. Sugeno FRBS model is defined by a set of N rules of the kind:

*R*ule i : If x_1 is A_1^i AND ...AND x_K is A_K^i THEN $z_i = f_i(x_1, \ldots, x_k)$

Where A_j^i are fuzzy constraints, x_j are numerical values corresponding to the linguistic variable j and z_i is a numerical value obtained by evaluation of function f_i. In our approach we used a 0-order Sugeno model characterized by using a constant function $f_i(x_1, \ldots, x_k) = c_i$. When, in a rule, we have multiple inputs its is necessary to aggregate them. For $k > 1$ inputs we have to use an operator and in the this implementation we used the min operator which is commonly used and is appropriate for this application.

$$w_i = \min\left(\mu_{A_1^i}(x_1), \ldots, \mu_{A_k^i}(x_k)\right) \tag{10}$$

The value of w_i is known as the "firing level"of rule i. The firing level of each rule is used to weight the function that in a way "translates" the *THEN* condition of each rule:

$$z = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i}$$

In Figure 5 a Sugeno model with two rules is presented.

4 Implementation details

4.1 Tabu Search

As mentioned before, since the goal of this paper was to evaluate the merit of an automatic procedure to tune tabu tenure, we have decided to implement a simple version of the TS. If more sophisticated features were introduce it could be difficult to access the merit of the contribution we are proposing.

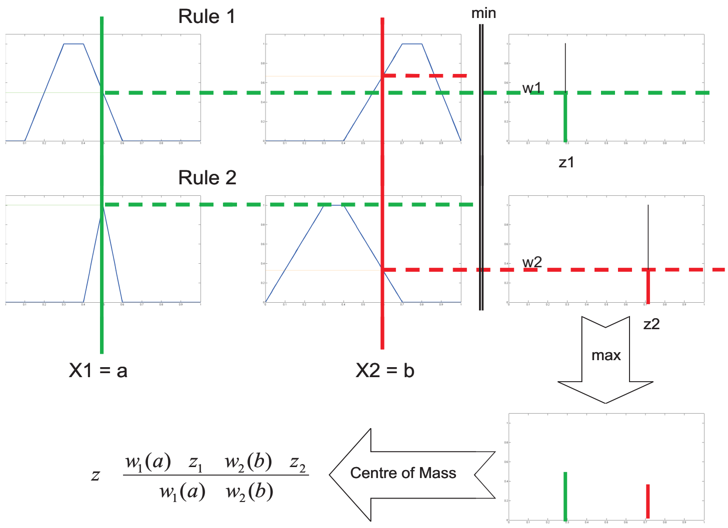

Figure 5: Sugeno FRBS with two rules

4.1.1 Solution Encoding

To encode the solution we used a vector structure with a dimension corresponding to the number of exams. The integer value saved in the i-component of the vector corresponds to the time slot of exam i. Given an example with 8 exams and 2 slots the next table represents a solution T_0 where, for instance, exams 1,2,3 and 4 are assign to slot 1.

T_0

Index of the vector (Exams)	1	2	3	4	5	6	7	8
Vector component (Time slot)	1	1	1	1	2	2	2	2

4.1.2 Initial Solution

In the application of TS to the exams timetabling problem we used a graph coloring heuristic, known as "*Saturation Degree*" [3] to find a starting solution. This option was based on a paper of Carter and Laport [12] where several heuristics were compared. This greedy heuristic mainly consists in ordering exams by the number of slots still available in increasing order and then assigning each exam to the first available slot. In presence of a tie the preference is given to the exam with more students.

4.1.3 Neighborhood

Two different neighborhoods were defined. A classical an elementary one corresponding, for a given timetable T_0, to all timetabling T_i differing from T_0 in the assignment of one exam alone. For this neighborhood a move consists in a period change for a given exam. For example, considering again the set of 8 exams and 2 slots, a possible neighbor of T_0 is timetable T_i where exam 3 change from slot (period) 1 to 2.

T_0

Exams	1	2	**3**	4	5	6	7	8
Time slot	1	1	**1**	1	2	2	2	2

T_i

Exams	1	2	**3**	4	5	6	7	8
Time slot	1	1	**2**	1	2	2	2	2

The second neighborhood used is based on *Kemp chains* introduced by Morgenstern [30a]. We define a neighborhood of timetable T_0, as the set of all timetables differing from T_0 only in the assignment of two groups of exams in two time slots. A move corresponds to a feasible interchanging of two sets of exams between two periods. For instance, given the example above of timetabeling T_0 we have a neighbor solution T_i where exams 1 and 2 in period 1 and exams 5,6 and 7 in period 2 interchanged periods. In order to preserve feasibility of the solution it may happen, in the limit, that a neighborhood solution consists solely in the interchange of periods between to groups of exams.

T_0

Exams	**1**	**2**	3	4	**5**	**6**	**7**	8
Timeslot	**1**	**1**	1	1	**2**	**2**	**2**	2

T_i

Exams	**1**	**2**	3	4	**5**	**6**	**7**	8
Timeslot	**2**	**2**	1	1	**1**	**1**	**1**	2

Figure 6 show the two kinds of moves just mentioned. A line connecting the index of two exams indicates that these exams share a number of enroled students. For both

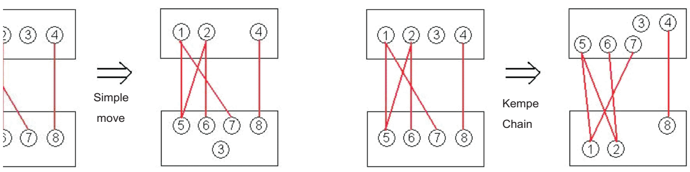

Figure 6: Simple and Kempe Chain moves

neighborhoods the all search space was inspected in each iteration. We used the objective function (7) to rank the points in the neighborhoods and chose the best available one (non tabu) to proceed.

4.1.4 Aspiration criteria

4.1.5 Memory

The memory management depends on the neighborhood that is considered. For the simple neighborhood it was recorder the index of the exam that was moved. As a consequence in a number of iterations equal to the tabu tenure we could not change the time slot of this exam. For the *kemp chains* neighborhood it would be to time and memory consuming to record the all chain of moves. Recording only all the exams that changed period would create an over restricting tabu list. In a few iterations it could happen that all the possible movements were tabu. So it was record a pair consisting in both the exam and the time slot of each exam that was involved in the chain of movements. In each iteration the tabu tenure of each element in the tabu list is decreased by one unit. The duration of the tabu status for each move is determined individually using a FRBS as is explained in the next section.

4.2 Fuzzy Rule Based System to manage the length of a tabu move

As it has been mentioned before the duration of a move inside the tabu list (tabu tenure) has a great impact in the performance of the TS. If the tabu tenure is low it can happen that in few iterations a local optima is revisited and so the algorithm enters in a loop.

On the other hand low values favor an intensification of the search in a region, which can be interesting if the region is promising and contains a good solution. In contrary, if the tabu tenure is high then the search space is diversified but a refined local search is not possible and good solutions can escape. This is why it is necessary, in general, to run the method repeatedly for the same instance, varying the value of the tabu tenure. This task is cumbersome if conducted manually. Even if this is implement automatically it is not straightforward to decide the range of values that should be tested for the tabu tenure, since it can varied greatly depending, for example, on the size of the instance. It has been already emphasized the importance of an automatic implementation of the TS, so it should be clear by now the relevance of a strategy that could handle, in an intelligent way, the choice of the tabu tenure for each element of the tabu list. Diversification and intensification of the search can be implicity performed through choices of respectively high and low values for the tabu tenure and should be balanced. One way to diversify the region of search is by avoiding movements that have been performed very often. On the other hand, since we do not register the solution itself but a characteristic associated with that solution, it is possible that solutions far apart share a common move. So if a move has not been active for a long period, and is activated now, its is possible that we are in a region far from the one where we were before when this move last entered in the tabu list. In this case there is no danger to allow for low values of the tabu tenure. To be able to distinguish these situations it was necessary to record some kind of historical data regarding the moves in the tabu list. Two features regarding the elements of the tabu list were registered, namely the *Frequency* and the *Inactivity*. The exact definition of these concepts depends on the neighborhood being considered. For the simpler neighborhood it was record for each exam i two arrays, namely:

Frequency(i)**-** the number of times that a move involving the change of the period of exam i was performed.

Inactivity(i)**-** the number of consecutive iterations for which there was no change in the period assign to exam i.

When using *kemp chains* neighborhood, for each pair defined by exam i and timeslot j, the above characteristics where defined in a similar way and registered in two matrices:

Frequency(i, j)**-** the number of times that a move involving the change of exam i to period j was performed.

Inactivity(i, j)**-** the number of consecutive iterations for which exam i remain assign to timeslot j.

In both cases *Inactivity* was set to zero, respectively, when exam i changed the time slot and when exam i was no longer assign to time slot j. The idea behind the FRBS is

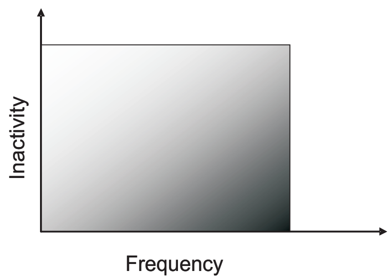

Figure 7: Tabu Tenure depending on Frequency and Inactivity

to emulate a behavior that will penalize moves, and will consequently set a high value for the tabu tenure, that present an high *Frequency* and low *Inactivity*. In opposition, moves with low *Frequency* and high *Inactivity* should have lower values for the tabu tenure. In Figure 7 we have an illustration on how the tabu tenure should be chosen according to the values of *Frequency* and *Inactivity*. Darker colors indicate higher values for the duration of a element in the tabu list. To emulate such behavior a FRBS was developed that could decide on the value tenure depending on these two inputs (*Frequency* and *Inactivity*). These two concepts where defined as Linguistic Variables (Section 3.3.2) with 3 linguist terms each - LOW, MEDIUM and HIGH, and fuzzyfied using a gaussian membership function as displayed in Figure 8.

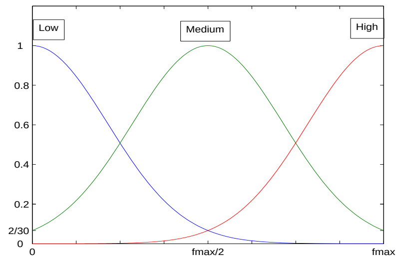

Figure 8: Frequency fuzzy membership function

Since we had two linguistic variables with 3 linguistic terms a total of 9 rules where defined:

Rule 1: **IF** *Frequency* is	LOW	**AND** *Inactivity* is HIGH	**THEN** $z_1 = 0.01\delta$
Rule 2: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is HIGH	**THEN** $z_2 = 0.04\delta$
Rule 3: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is HIGH	**THEN** $z_3 = 0.07\delta$
Rule 4: **IF** *Frequency* is	LOW	**AND** *Inactivity* is MEDIUM	**THEN** $z_4 = 0.02\delta$
Rule 5: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is MEDIUM	**THEN** $z_5 = 0.05\delta$
Rule 6: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is MEDIUM	**THEN** $z_6 = 0.08\delta$
Rule 7: **IF** *Frequency* is	LOW	**AND** *Inactivity* is LOW	**THEN** $z_7 = 0.03\delta$
Rule 8: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is LOW	**THEN** $z_8 = 0.06\delta$
Rule 9: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is LOW	**THEN** $z_9 = 0.1\delta$

The value of δ is defined by a function that depends on the caracteristics of the problem instance and the type of neighborhood. When using the simple neighborhood then the function to generate δ depends on the number of exams and the density of the conflict matrix.

$$\delta = \text{Number of exams} * (1 - \text{ Density of conflicting matix})^3$$

For the Kempe Chains the function also considers the number of periods.

$$\delta = \text{Number of exams} * \text{Number of periods} * (1 - \text{ Density of conflicting matix})^3$$

A 0-order Sugeno system was implemented (Subsection 3.3.4). Given a value for the *Frequency* x_F the membership value for **LOW**, **MEDIUM** and **HIGH** was obtain yielding the values $\mu_L^F(x_F)$, $\mu_M^F(x_F)$, $\mu_H^F(x_F)$ respectively. The same for inactivity, giving $\mu_L^I(x_I)$, $\mu_M^I(x_I)$, $\mu_H^I(x_I)$. The firing level for each rule is defined according to (10). For instance, in Rule 7 the firing level of the rule will be $w_7(x_F, x_I) = \min\{\mu_M^F(x_F), \mu_L^I(x_I)\}$. So for each rule we will obtain a firing level of value $w_i(x_F, x_I)$ and the final output will be given by:

$$z = \frac{\sum_{i=1}^{7} w_i(x_F, x_I) z_i}{\sum_{i=1}^{7} w_i(x_F, x_I)} \tag{11}$$

This value was rounded to an integer value to define the tabu tenure.

5 Computational results

To test if the FRBS improves the performance of the TS we have tested our algorithm on a test bed of real problems available on a online repository created by Michael Carter[1] and often used in the literature of exams timetabling. We have implemented the TS in Matlab R2007b and performed the computational experiments on a Pentium Intel Core2 Duo T9400 with 2.53GHz and 3 Gb of memory. Figure 9 shows the interface of the

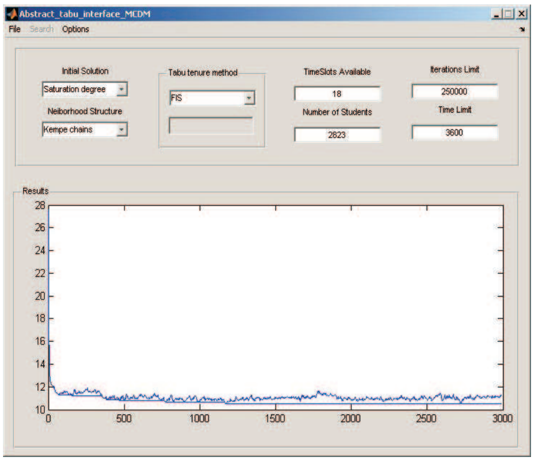

Figure 9: Matlab Interface

program. The user could define choice of the initial solution, the neighborhood to be used, the strategy for handle the memory and the stopping criteria. A graphic shows the evolution of the algorithm in regard to the value of the objective function.

We have analyzed the results in two ways. First we wanted to compare the performance of the algorithm using FRBS and using fixed values for the tabu tenure. We wanted to use 3 different values that could be consider as low, medium and high. For that we gather information during the run of the FRBS and took the lowest, medium and the highest provided value for the tabu tenure. The stopping criteria was a limit of one hour for computational time or 250000 iterations. This last rule was never effective, the stopping condition was always the computational time. In the next table is presented, for each instance, the value of the objective function using the FRBS in column FIS and followed respectively by the results when using the lowest, medium and highest value for the tabu tenure.

[1] ftp://ftp.mie.utoronto.ca/pub/carter/testprob

Instance	FIS	Lowest tenure value	Medium tenure value	Highest tenure value
car-f-92	4,57	4,57	4,57	4,57
car-s-91	5,46	5,46	5,46	5,46
ear-f-83	33,50	36,41	34,81	36,52
hec-s-92	10,52	11,37	10,43	10,48
kfu-s-93	14,05	14,03	14,11	14,11
rye-s-93	9,11	9,11	9,12	9,11
sta-f-83	157,29	157,30	157,30	157,29
tre-s-92	8,71	9,00	9,26	9,26
uta-s-92	3,71	3,71	3,73	3,73
ute-s-92	25,18	26,06	25,39	24,99
yor-f-83	39,08	37,25	37,06	39,37

To interpreter these results we have analysed the number of times that each strategy yield the best, the second best, the third best and the worse result. Figure 10 shows the

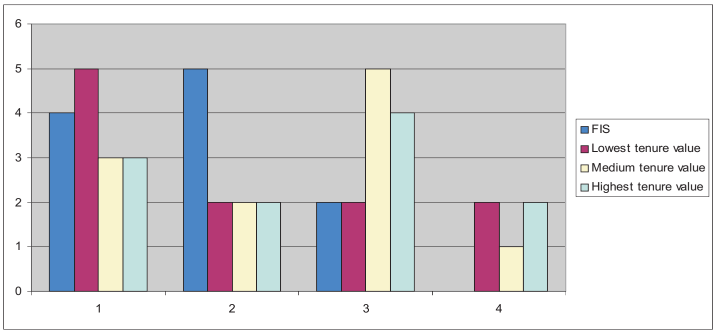

Figure 10: Comparison of methods

classification of each strategy and we can see FIS was never the worse strategy and most of the times the best or the second best one.

The second goal was to compare our implementation of the TS with the FRBS with other methods in the literature using the same objective function (7) and the same problems. We used the papers that presented the best known value for at least one problem from the test bed. Next table shows the results obtained when we run the TS with the stopping criteria of time less than 1 hour.

	FIS	Caramia	Y. &P.	Burke	C. & T.
car-f-92	4,57	6.6	**4.5**	4.6	5.4
car-s-91	5,46	6.0	3.93	**4.0**	4.4
ear-f-83	33,50	**29.3**	33.7	32.8	34.8
hec-s-92	10,52	**9.2**	10.83	10.0	10.8
kfu-s-93	14,05	13.8	13.82	**13.0**	14.1
sta-f-83	157,29	158.2	158.35	159.9	**134.9**
tre-s-92	8,71	9.4	7.92	**7.9**	8.7
uta-s-92	3,71	3.5	3.14	**3.2**	-
ute-s-92	25,18	**24.4**	25.39	24.8	25.4
yor-f-83	39,08	**36.2**	36.35	37.28	37.5

The columns of the table represent:

FIS - Fuzzy Inference Rule Based System

Caramia - Caramia et. al.(2001)[8]

Y. &P. - Yang & Petrovic(2006) [45]

Burke - Burke et. al.(2006)[4]

C. & T - Casey & Thompson(2003) [13]

The best result is highlighted in bold. We can see that although we never obtained the best solution in column FIS the value was very closed to the best one giving an average gap of 15%. We must take in consideration that we have implemented a plain TS, and limiting the computational time to 1 hour much less than the computational time of the other methods. In some cases this time was around 12 hours.

6 Conclusions

In this paper we propose the implementation of a Fuzzy Rule Based System to chose the tabu tenure for each element of the tabu list in a Tabu Search for the exams timetabling problem. The FRBS has as input the Frequency and Inactivity of the tabu and was design in order to balance the possibility of a diversified search without compromising the intensification. We show, using a test bed of real problems that this methodology performs well when compared to strategies with a fixed value for the tabu tenure. Comparing the performance with other methods in the literature, and taking in consideration that we implemented a simple and limited version of the TS we can state that this methodology is a good alternative to the usual process of tuning the tabu tenure. The idea behind this application can be extended to other applications of the TS.

References

[1] Abramson D. A., Dang H. and Krisnamoorthy M. (1999), "Simulated Annealing Cooling Schedules for the School Timetabling Problem", Afia-Pacific Journal of Operational Research, Vol. 16, pp 1-22.

[2] Awad R., Chinneck J., (1998) "Proctor assignment at Carleton University". Interfaces vol. 28, no. 2, pp 58-71.

[3] Brelaz D (1979). New Methods to Colour the Vertices of a Graph. Communication of the ACM 22(4): 251-256.

[4] Burke EK , Eckersley AJ , McCollum B , Petrovic S, Qu R (2006). Hybrid variable neighbourhood approaches to university exam timetabling. Technical Report NOTTCS-TR-2006-2, School of CSiT, University of Nottingham.

[5] Burke EK, Kingston JH, de Werra D (2004). Applications to timetabling. In: J. Gross and J. Yellen (eds.) The Handbook of Graph Theory, Chapman Hall/CRC Press, 2004, 445-474.

[6] Burke, E. K. and J. P. Newall (2003). Enhancing timetable solutions with local search methods. Practice and Theory of Automated Timetabling IV, Springer-Verlag.

[7] Caprara A., Fischetti M., Guida P. L., Monaci M., Sacco G., Toth P., (2001), "Solution of real-world train timetabling problems", Proceedings of the 34th Annual Hawaii International Conference on Systems Sciences, pp1057-1066.

[8] Caramia M, DellOlmo P, Italiano GF (2001). New algorithms for examination timetabling. In: S. Naher and D. Wagner (eds): Algorithm Engineering 4th International Workshop, Proceedings WAE 2000. Springer Lecture Notes in Computer Science, vol. 1982, 230-241.

[10] Carter, M. W. and G. Laporte (1996). Recent Developments in Practical Examination Timetabling. First International Conference on Practice and Theory of Automated Timetabling.

[11] Carter, M. W., G. Laporte, et al. (1994). "A General Examination Scheduling System." Interfaces 24(3): 109-120.

[12] Carter, M. W., G. Laporte, et al. (1996). "Examination timetabling: Algorithmic strategies and applications." Journal of Operational Research Society 47(3): 373-383.

[13] Casey S, Thompson J (2003). GRASPing the examination scheduling problem. In: E.K. Burke and P. De Causmaecker (eds). (2003). Practice and Theory of Automated Timetabling: Selected Papers from the 4th International Conference. Springer Lecture Notes in Computer Science, vol. 2740. 232-244.

[14] Chun, H. W., H. C. Chan, et al. (2000). Nurse Rostering at the Hospital Authority of Hong Kong. Twelfth Conference on Innovative Applications of Artificial Intelligence, Austin.

[15] Colorni A., Dorigo M. and Maniezzo V. (1998) "Metaheuristic for high-school timetabling" Computational Optimization and Applications, Vol. 9, Kluwer Acad. Publ., Dodrecht, NL, pp 275-298.

[16] Corr, P. H., B. McCollum, et al. (2006). A New Neural Network Based Construction Heuristic for the Examination Timetabling Problem. Parallel Problem Solving from Nature - PPSN IX, Springer.

[17] Desroches, S., G. Laporte, et al. (1978). "HOREX: A computer program for hte construction of examination." INFOR 16(3): 294-298.

[18] Dowsland, K. A. and J. M. Thompson (2005). "Ant colony optimisation for the examination scheduling problem. ." Journal of the Operational Research Society 56: 426-438.

[19] Easton, K., G. Nemhauser, et al. (2004). Sports Scheduling. Handbook of Scheduling: Algorithms, Models, and Performance Analysis. J. Y.-T. Leung, CRC Press.

[20] Erben, W. (2001). A grouping genetic algorithm for graph colouring and exam timetabling. The Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium, Springer-Verlag.

[21] Gaspero, L. D. (2002). Recolour, shake and kick: A recipe for the examination timetabling problem. 4th International Conference on Practice and Theory of Automated Timetabling, Gent, Belgium.

[22] Gaspero, L. D. and A. Schaerf (2000). Tabu Search Techniques for Examination Timetabling. Practice and Theory of Automated Timetabling III, Third International Conference, Konstanz, Germany, Springer

[23] Gaspero, L. D. and A. Schaerf (2001). Tabu Search Techniques for Examination Timetabling. Practice and Theory of Automated Timetabling III, Third International Conference, Konstanz, Germany, Springer

[24] Glover F (1986), Future paths for interger programming and links to artificial intelligence, Computer & Operation Research 13(5): 533-549.

[25] Glover F, Laguna M (1997), Tabu Search, Kluwer Academic Publisher.

[27] Hansen M. P., "Planning of high-school examinations in Denmark2, European Journal of Oprational Research, vol. 87, pp. 519-534.

[28] Isaai M. T., Singh M. G., (2001). "Hybrid applications of constraint satisfaction and meta-heuristics to railway timetabling: a comparative study". IEEE Transactions on Systems, Man. and Cybernetics, Part C: Applications and Reviews, Volume: 31 Issue: 1, pp 87-95.

[26] Kendall, G. and Hussin N. M. (2005). An investigation of a tabu search based hyper-heuristic for examination timetabling. Multidisciplinary Scheduling: Theory and Applications, Springer.

[29] Kwan, R. (2004). Bus and Train Driver Scheduling. Handbook of Scheduling: Algorithms, Models, and Performance Analysis. J. Y.-T. Leung, CRC Press.

[30] Merlot, L. T. G., N. Boland, et al. (2003). A hybrid algorithm for the examination timetabling problem. Practice and Theory of Automated Timetabling: Selected Papers from the 4th International Conference, Springer Lectures Notes in Computer Science.

[30a] Morgenstern, C. (1989). Algorithms for general graph coloring. Department of Computer Science, University of New Mexico. PhD Thesys.

[31] Pacquete, L. and C. M. Fonseca (2001). A study of examination timetabling with multi-objective evolutionary algorithm. The Proceedings of the 4th Meta-heuristics International Conference (MIC 2001).

[32] Petrovic, S. and Y. Bykov (2003). A Multiobjective Optimisation Technique for Exam Timetabling Base on Trajectories. The Practice and Theory of Automated Timetabling IV, Springer-Verlag.

[33] Qi, X., J. Yang, et al. (2004). Scheduling Problems in the Airline Industry. Handbook of Scheduling: Algorithms, Models, and Performance Analysis. J. Y.-T. Leung, CRC Press.

[34] Qu, R., E. Burke, et al. (2006). A Survey of Search Methodologies and Automated Approaches for Examination Timetabling. Nottingham, University of Nottingham.

[35] Ribeiro Filho G. and Lorena L. A. N., (2001). "A Construtive Evolutionary approach to school timetabling". In Applications of Evolutionary Computing. Boers, E. J. W., Gottlieb J., Lanzi P. L., Smith R. E., Cagnoni S., Hart E., Raidl G. R., Tijink H., (Eds.) - Springer Lectures Notes in Computer Science, vol. 2037, pp 130-139.

[36] Schimmelpfeng, K. and S. Helber (2007). " Application of a real-world university-course timetabling model solved by integer programming." OR Spectrum 29(4): 783-803.

[37] Sugeno, M, Kang G (1988). "Structure identification of fuzzy model." Fuzzy Sets and Systems 28: 15-33.

[38] Salwani Addullah (2006) Ph.D. Thesis, University of Nottingham, UK.

[39] Schaerf A. and Schaerf M. (1996). "Local search techniques fot high school timetabling", Proccedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling PATAT 2000, pp 128-147, Konstanz, 2000, The Practice and Theory of Automated Timetabling: Volume 3, E. K. Burke and W. Erben (eds) pp 242-253.

[40] Takagi T , Sugeno M (1985). "Fuzzy identification of fuzzy of systems and its applications to modeling and control." IEEE Transactions on Systems, Man, Cybernetics 15: 116-132.

[41] Thompson, J. M. and K. A. Dowsland (1996). "Variants of simulated annealing for the examination timetabling problem." Annals of Operations Research 63: 105-128.

[42] Thompson, J. M. and K. A. Dowsland (1998). "A Robust Simulated Annealing based Examination Timetabling System." Computer & Operation Research 25(7): 637-648.

[43] Trick M. A. (2000). "A Schedule-then-Break approach to sports timetabling", Proccedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling PATAT 2000, pp 128-147, Konstanz, 2000, The Practice and Theory of Automated Timetabling: Volume 3, E. K. Burke and W. Erben (eds) pp 242-253.

[44] White, G. M. and B. S. Xie (2001). Examination timetables and tabu search with longer-term memory. Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference, Springer.

[45] Yang Y, Petrovic S (2004). A Novel similarity measure for heuristic selection in examination timetabling. In: E.K. Burke and M. Trick (eds). (2005). Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling. Springer Lecture Notes in Computer Science, vol. 3616. 377-396.