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

Tiago Cardal Pais * P. Amaral †

3rd July 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 with some parameter that can affect the performance of the algorithm and must be tuned. One of such is tabu tenure which is much problem dependent. Our goal is to have a automatic process of managing the memory which is important for practical applications. We can in general consider that the degree of expertise of the users (university staff for instance) is low. To automatically manage the tabu tenure we employ techniques used in fuzzy decision expert systems. We developed a fuzzy inference rule based system (FIRBS) 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. We have implemented the simplest form of a tabu search in order to evaluate this new feature. Computational results show that using a FIRBS improves the performance of Tabu Search.

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

1 The examination timetabling problem

Problems related to timetabling are present in daily life. Solving timetabling problems is a crucial task and affect many institutions and services like hospital, transportation enterprizes, educational establishments, among many others. This problems have been an object of increasingly interest since the 1960s. Mainly in the field of Operations Research and Artificial Intelligence many interesting proposals have been presented. A general definition of timetabling was given by Burke, Kingston and Werra [3].

" A timetabling problem is a problem with four parameters, T a finite set of times, R a 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 problems. This problems have attracted a considerable interests absolutely justified by

*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

its importance and relevance in educational success. This task together with classes timetabling are one of the most hard and time consuming tasks in scholar management. In the last years we have noticed a tendency towards the flexibility of curricula which have introduced an additional source of difficulty in an already hard optimization problem. This 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. A possible integer formulation for this problem is the following:
Integer Formulation

$$\begin{aligned}
\text{Let } c_{ij} &= \text{number of students enroled in course } i \text{ and } j && (1)\\
A_{ij} &= \begin{cases} 1 & \text{if } c_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} && (2)\\
K &= \text{number of courses} && (3)\\
P &= \text{number of slots} && (4)\\
M &= \text{number of students} && (5)
\end{aligned}$$

and consider the following variables

$$\begin{aligned}
x_i &= \text{slot to which the exam is assigned } i = 1, \ldots, P && (6)\\
y_{ij} &= \begin{cases} 16 & \text{if } |x_i - x_j| = 1 \\ 8 & \text{if } |x_i - x_j| = 2 \\ 4 & \text{if } |x_i - x_j| = 3 \\ 2 & \text{if } |x_i - x_j| = 4 \\ 1 & \text{if } |x_i - x_j| = 5 \\ 0 & \text{otherwise .} \end{cases} && (7)
\end{aligned}$$

The formulation is:

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

Where M_0 is an arbitrarily large number. This formulation was adapted from [9]. 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 [3], [2], [4], [5], [11]. Tabu search is one of such methods that have been applied to solve these problems.

2 Tabu Search

Tabu search [6], [7] 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

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 the 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. We proceed to the next iteration from this current point. There are many interesting additional refinements that can greatly increase the performance of TS.

2.1 Implementation details

In the application of TS to the exams timetabling problem we used a graph coloring heuristic, known as "*saturation degree*" [1] to find a starting solution. Two different neighborhoods were defined. A classical an elementary one that corresponds, 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 a set of 8 exams and tree slots a neighbor of T_0 is a timetable T_i where exam 3 change from slot (period) 3 to 2.

T_0

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

T_i

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

The second neighborhood used is based on *Kemp chains*. 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 2 and 6 in period 2 and exams 3 and 7 in period 3 interchanged periods.

T_0

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

T_i

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

For both neighborhoods the all search space was inspected in each iteration. We used the objective function (8) to rank the points in the neighborhoods and chose the best available (non tabu) one. The memory management depended on the neighborhood. 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 consuming to record the all chain of moves. Recording only the exam that changed period would create an over restricting tabu list. So it was record both the exam and the time slot of each exam that was involved in the chain of movements.

3 Fuzzy Inference System for the length of Tabu List

We have incorporated a method to automatically managed the tabu tenure of each member of the tabu list. This approach is based on a fuzzy inference rule based system. Historical information regarding moves is record and used in a twofold way. 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 exam i period was performed.

Inactivity(i)**-** the number of consecutive iterations for which there was no change in the time slot 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 time slot and when exam i was no longer assign to timeslot j. This data was used in a Fuzzy Inference Rule Based System (FIRBS). The goal of a FIRBS is to emulate the behavior of human reasoning. A rule based system can generically be defined by a set or rules of the type

Rule k : IF X THEN Y.

A FIRBS uses fuzzy sets instead of crisp sets. In general linguistic variables are used as input (and eventually as output). In this approach a fuzzyfication of linguistic variables *Frequency* and *Inactivity* was perform using three categories - LOW , MEDIUM, HIGH - and using an gaussian membership function as displayed in Figure 1.

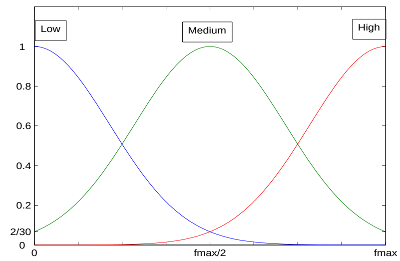

Figure 1: Frequency fuzzy membership function

We used a 0-order Sugeno (or TSK) FRBS [10], [8], that uses rules such as:

Rule k : IF X_1 is A_1^k AND X_2 is A_2^k AND ... X_r is A_r^k THEN $z_k = c_k$.

To handle the **AND** relation the usual **MIN** operator was used.

$$\mu_k = \min\{\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \ldots, \mu_{A_r^k}(x_r)\} \tag{10}$$

where $\mu_{A_i^k}(x)$ is the membership value of x for the category A_i^k. The main idea defining the rules consists in penalizing moves that were often and recently called. If for instance a move has frequently and recently been performed we assign a long tenure for this element of the tabu list. In opposition, to a move that was seldom and not recently performed will be given a low tabu tenure value. Since we had two linguistic variables we defined a total of 9 rules.

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 problem attributes. Given a value for the *Frequency* x_F the membership value for **LOW**, **MEDIUM** and **HIGH** was obtain yielding the values $\mu_L^F(x_F)$, $\mu_M^F(x_F)$, $\mu_H^F(x_F)$. 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.

4 Computational results

To test if the FIRBS 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 compared our results with those obtained from other authors using the same objective function (8) and the same problems. Next table shows the results obtained when we run the TS with the stopping criteria of time less than 1 hour.

	FIS	L.T.V.	M.T.V.	H.T.V.	Caramia	Y. &P.	Burke	C. & T.
car-f-92	4,57	4,57	4,57	4,57	6.6	4.5	4.6	5.4
car-s-91	5,46	5,46	5,46	5,46	6.0	3.93	4.0	4.4
ear-f-83	33,50	36,41	34,81	36,52	29.3	33.7	32.8	34.8
hec-s-92	10,52	11,37	10,43	10,48	9.2	10.83	10.0	10.8
kfu-s-93	14,05	14,03	14,11	14,11	13.8	13.82	13.0	14.1
sta-f-83	157,29	157,30	157,30	157,29	158.2	158.35	159.9	134.9
tre-s-92	8,71	9,00	9,26	9,26	9.4	7.92	7.9	8.7
uta-s-92	3,71	3,71	3,73	3,73	3.5	3.14	3.2	-
ute-s-92	25,18	26,06	25,39	24,99	24.4	25.39	24.8	25.4
yor-f-83	39,08	37,25	37,06	39,37	36.2	36.35	37.28	37.5

The columns of the table represent:

FIS-	Fuzzy Inference Rule Based System	L.T.V.-	Lowest tenure value
M.T.V.-	Medium tenure value	H.T.V.-	Highest tenure value
Caramia	Caramia et. al.(2001)[4]	Y. &P.-	Yang & Petrovic(2006) [11]
Burke-	Burke et. al.(2006)[2]	C. & T.-	Casey & Thompson(2003) [5]

To resume the comparison between the values of columns 1 to 4 we present the following histogram. The y-axis represents the number of times that one approach obtained the k^{th} best result and the x-axis represents the k best result. Observing the histogram we see

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

that when we use the FIRBS to choose the tabu tenure we never obtain the worse results and only a couple of times third best results was obtained. Therefore using the FIRBS in a TS approach provides a stable and suitable algorithm for automatic approaches.

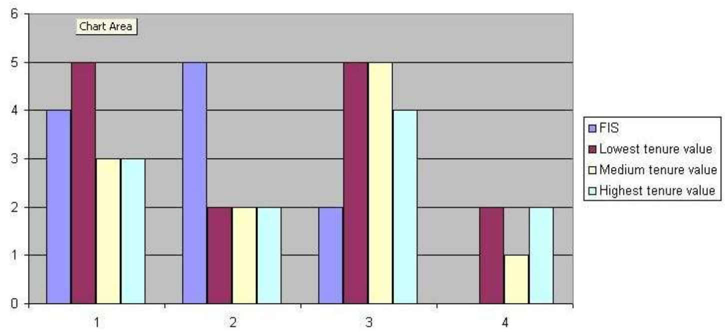

Figure 2: Comparison of methods

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

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

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

[10] 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.

[11] 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.