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 drawbacks 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 an 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 attempted 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 enterprises, 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 set of resources, M, a finite set of meetings: and C, a finite set of constraints. The problem is to assign times and resources 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 prob-
lem. 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**

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

and consider the following variables

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

The formulation is:

\[
\min f = \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} c_{ij} y_{ij}}{2M}
\]

subject to 
\(|x_i - x_j| \geq 1 - (1 - A_{ij})M_0\) for \( i, j = 1, \ldots, K, i \neq j \)

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 unsuc-
cessful 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.

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 timetabling $T_0$ 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.

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

$$\text{Rule } k : \text{IF } X \text{ THEN } Y.$$ 

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.

![Frequency fuzzy membership function](image)

Figure 1: Frequency fuzzy membership function

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

$$\text{Rule } k : \text{IF } X_1 \text{ is } A^k_1 \text{ AND } X_2 \text{ is } A^k_2 \text{ AND } \ldots X_r \text{ is } A^k_r \text{ THEN } z_k = c_k.$$ 

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

$$\mu_k = \min\{\mu_{A^k_1}(x_1), \mu_{A^k_2}(x_2), \ldots, \mu_{A^k_r}(x_r)\} \quad (10)$$

where $\mu_{A^k}(x)$ is the membership value of $x$ for the category $A^k$. 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.
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 an 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>car-92</td>
<td>4.51</td>
<td>4.51</td>
<td>4.51</td>
<td>6.6</td>
<td>4.5</td>
<td>4.6</td>
<td>5.4</td>
</tr>
<tr>
<td>car-91</td>
<td>5.46</td>
<td>5.46</td>
<td>5.46</td>
<td>6.0</td>
<td>5.93</td>
<td>4.9</td>
<td>4.4</td>
</tr>
<tr>
<td>hec-92</td>
<td>33.93</td>
<td>34.81</td>
<td>36.52</td>
<td>29.3</td>
<td>33.7</td>
<td>32.8</td>
<td>34.8</td>
</tr>
<tr>
<td>kfu-93</td>
<td>10.52</td>
<td>11.37</td>
<td>10.43</td>
<td>9.2</td>
<td>10.83</td>
<td>10.0</td>
<td>10.8</td>
</tr>
<tr>
<td>sta-93</td>
<td>14.05</td>
<td>14.05</td>
<td>14.11</td>
<td>13.8</td>
<td>13.82</td>
<td>13.0</td>
<td>14.1</td>
</tr>
<tr>
<td>tre-92</td>
<td>8.71</td>
<td>9.00</td>
<td>9.26</td>
<td>9.26</td>
<td>7.92</td>
<td>7.9</td>
<td>8.7</td>
</tr>
<tr>
<td>uta-92</td>
<td>3.71</td>
<td>3.71</td>
<td>3.73</td>
<td>3.73</td>
<td>3.14</td>
<td>3.2</td>
<td>-</td>
</tr>
<tr>
<td>yor-93</td>
<td>39.08</td>
<td>37.25</td>
<td>37.06</td>
<td>39.37</td>
<td>36.35</td>
<td>37.28</td>
<td>37.5</td>
</tr>
</tbody>
</table>

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]
- **Burke**: Burke et. al. (2006)[2]

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