MTISD 2008 – Methods, Models and Information Technologies for Decision Support Systems
Università del Salento, Lecce, 18-20 September 2008

Using weight aggregation in tabu search for multiobjective exams timetabling problem

Tiago Cardal Pais
Uninova, Portugal
tpp@uninova.pt
Paula Amaral
Dep. Math., FCT, UNL, Portugal.
paca@fct.unl.pt

Abstract

Exams timetabling is a difficult task in many educational institutions. We can distinct two major sets of constraints when defining exams timetabling problems, categorized in soft and hard constraints. Guaranteing that any student as a non overlapping exams schedule and that necessary requirements like rooms and teacher are available are consider hard constraints. An evenly distributed schedule, a short duration of the overall exams period can be regarded as soft constraints. To handle soft constraints under the hard constraints verification we adopted a multiobjective optimization approach. This problem is NP-hard for which we have developed an heuristic tabu search method to find a solution. Tabu search comprises an iterative local search defined as a neighborhood inspection of a certain point in the search space. To find an improved solution we have to evaluate points in this neighborhood which can be considered a multiple attribute decision problem. In this context we have used multicriteria methods in order to rank the solutions.

1 Introduction

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, educational establishments, among many others. This problems have been an object of increasingly interest in the four last decades. In particular, regarding exams timetabling problems, the interests is absolutely justified by its importance and relevance in educational success. In general we can say that an exam timetabling problem consists of finding a feasible schedule for each student in the sense that no two exams overlap and such that other requirements such as rooms and staff are fulfill. But the quality of a schedule depends also on other factors such as the spreading of exams, allowing for more time between consecutive exams and the minimization of the examination period. Although specifications of the problems can differ, essentially we have the following input data.

$$N = \text{number of exams.} \quad (1)$$
$$c_{ij} = \text{number of students enroled in course } i \text{ and } j \quad (2)$$
$$K = \text{number of courses} \quad (3)$$
$$P = \text{number of slots} \quad (4)$$
$$M = \text{number of students} \quad (5)$$

We encoded the solution using a vector of variables $T = (t_i), i = 1, \ldots, N$, such that t_i represents the timeslot assign to exam i, and a set of variables $d_{t_i}, i = 1, \ldots, N$ which represents the day where exam t_i takes place. As in [2] we have consider the following four objectives:
- The number of conflicts where students have exams in adjacent periods of the same day

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.adjs(t_i, t_j) \text{ where } adjs(t_i, t_j) = \begin{cases} 1 & \text{if } (|t_i - t_j| = 1) \wedge (d_{t_i} = d_{t_j}) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

- The number of conflicts where students have two or more exams in the same day.

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}.sday(t_i,t_j) \text{ where } sday(t_i,t_j) = \begin{cases} 1 & \text{if } d_{t_i} = d_{t_j} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

- The number of conflicts where students have exams in adjacent days.

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}.adjd(t_i,t_j) \text{ where } adjd(t_i,t_j) = \begin{cases} 1 & \text{if } |d_{t_i} - d_{t_j}| = 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

- The number of conflicts where students have exams in overnight adjacent periods

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}.ovnt(t_i,t_j) \text{ where } ovnt(t_i,t_j) = \begin{cases} 1 & \text{if } (|t_i - t_j| = 1) \wedge (|d_{t_i} - d_{t_j}| = 1) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

We consider a single set of hard constaints, to guarantee that no student has the same exam in the same slot:

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}.clash(t_i,t_j) = 0 \text{ where } clash(t_i,t_j) = \begin{cases} 1 & \text{if } t_i = t_j \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

This problem is known to be NP-hard and we have implemented a Tabu Search (TS) to obtain good feasible solutions.

2 Tabu search

Tabu search [3], [4] 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 - Enables to override the tabu.

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. We add the move (or solution) 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. 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. The second neighborhood used is based on *Kemp chains*. We define a neighborhood of timetable T_0, as the set of all the timetables differing from T_0 only in the assignment of two groups of exams in two time slots. Also a Inference Rule Based (IRB) system was developed to manage the size of the tabu list, to increase the automatization of this procedure. For the evaluation of the solutions we use some strategies borrowed from multiple criteria decision making.

3 Using weight aggregation for evaluation and selection

Once a set of points where generated, as a neighborhood of a current solution, it was necessary to evaluate the candidates in order to find an eligible solution. In a problem with only one objective, the value of the objective function is often used to rank the solutions. In this case we had a multiobjective problem and we have chosen to maintain the problem as such, instead of transforming it in a single objective problem using functions aggregation. The reason to do so was based on an attempt to do not prematurely condition the problem, allowing for a broader inspection of solutions in a sort of diversification strategy.
So for a given set of points, $T_1, T_2, \ldots, T_r$ we have a set of their corresponding values for the above mentioned objectives functions f_1, f_2, f_3, f_4. Identifying the points as alternatives and the value of the objective functions as attributes that characterize the alternatives, we are facing a multiple attribute decision making (MADM) problem.
We performed a normalization of the data in order to be able to compare the attributes, defined by a matrix $\mathbf{X} = (x_{ij})$, where $0 \le x_{ij} \le 1$. The ranking of the points T_i was done using a Compromise Ratio (CR) methodology [5]. The compromise ratio is developed based on the concept that the best alternative should be as close s possible to the ideal solution a^+ and as far as possible to the negative-ideal solution a^-, which in this case are consider to be a four dimension vector of ones and zeros respectively. Since the attributes may have different degrees of importance for the decision maker a component wise weighting of matrix $\mathbf{X} = (x_{ij})$ was performed,

$$v_{ij} = x_{ij} \times w_{ij}. \tag{11}$$

For each point T_i we need to compute the distances to the ideal and negative-ideal point, respectively

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m} (a_j^+ - v_{ij})^p} \qquad , \forall i = 1, ..., n$$

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m} (v_{ij} - a_j^-)^p} \qquad , \forall i = 1, ..., n.$$

Given these distances we can use the following parameterized ratio

$$\xi_p(T_i) = \theta \times \frac{D_{p^-}(T^+) - D_p^+(T_i)}{D_{p^-}(T^+) - D_{p^+}(T^+)} + (1-\theta) \times \frac{D_p^-(T_i) - D_{p^-}(T^-)}{D_{p^+}(T^-) - D_{p^-}(T^-)}$$

where

$$\begin{cases} D_{p^-}(T^+) = \max\limits_{i \in \{1,\ldots,n\}} \{D_p^+(T_i)\} \\ D_{p^+}(T^+) = \min\limits_{i \in \{1,\ldots,n\}} \{D_p^+(T_i)\} \\ D_{p^+}(T^-) = \max\limits_{i \in \{1,\ldots,n\}} \{D_p^-(T_i)\} \\ D_{p^-}(T^-) = \min\limits_{i \in \{1,\ldots,n\}} \{D_p^-(T_i)\} \end{cases}$$

in order to rank the alternatives. However we have modified the above mention procedure by using in (11) weight generating functions g_j instead of constant weights, yielding

$$W_{ij}(\mathbf{x}_i) = \frac{g_j(x_{ij})}{\sum\limits_{t=1}^{m} g_t(x_{it})}.$$

The gain is that we can better model preferences as well as the behavior of the decision maker than if we simply use constant weights. For instance, consider the classical situation of buying a car. For that purpose, we are evaluating the car regarding its price and comfort. If the car is expensive it is expected to be also very comfortable. However, if the car is not expensive there is no such expectation. In this situation the weight of each criteria is related with the criteria satisfaction. Using a weighting function it is possible to model this situation.

Mixture operators, in the context of aggregation operators where introduced in [8] and [9]. In [6], [7] we can find some interesting applications. We know that the operator we use is not monotonic for all weighting functions. In order to guarantee the monotonicity of the operator ξ_p we obtained a condition similar to those presented in [8], [9] and [10].

We used the following weight generating function:

$$g_j(x_{ij}) = (a_j - d_j) \times x_{ij}^p + d_j$$

Where a_j represents the importance (weight) of attribute j when the attribute satisfaction is maximum and it value belongs to the unit interval, and d_j represent the importance (weight) of attribute j when the attribute satisfaction is minimum. However d_j belongs to the interval $[\text{lower\_bound}, a]$. The lower_bound value is derived from the same condition referred above.

We performed some computational experiments on a test set of real problems available on an online repository of exams timetabling problems. The numerical results proved that this approach successfully enhances the features of the Tabu Search.

References

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

[2] Bykov Y. (2003), Time-Predefined and Trajectory-Based Search: Single and Multiobjective Approches to Exam Timetabling, The School of Computer Science and Information Technology, The University of Nottingham.

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

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

[5] Li D.-F. (2007). Compromise ratio method for fuzzy multi-attribute group decision making. Applied Soft Computing 7(3): 43 - 58.

[6] Marques Pereira R. A., Ribeiro R. A. (2003), Aggregation with generalized mixture operators using weighting functions, Fuzzy Sets and Systems 137(1): 43 - 58.

[7] Marques Pereira R. A., Ribeiro R. A. (2003), Generalized mixture operators using weighting functions: a comparative study with WA and OWA, European Journal of Operational Research, 145: 329-342.

[8] Marques Pereira R. A., Pasi G. (1999), On non-monotonic aggregation: mixture operators, Proc. Intl. Workshop EUROFUSE 1999, Intl. Conference on Soft and Intelligent Computing SIC, Technical University of Budapest,Hungary: 513-517,

[9] Marques Pereira R. A. (2000), The orness of mixture operators: the exponential case, Proc. 8th Intl. Conf. on Information Processing and Management of Uncertainty, IPMU 2000, Technical University of Madrid, Spain, 2: 974-978.

[10] Mesiar R. Spirkova J. Vavrikova L. (2008), Weighted aggregation operators based on minimization, Information Sciences, 178(4): 1133-1140.