Weight aggregation in a multiobjective approach for exams timetabling problems

Tiago Cardal Pais
School of Computer Science, University of Nottingham, Nottingham UK
txp@cs.nott.ac.uk
Paula Amaral
Dep. Math., FCT, UNL, Portugal.
paca@fct.unl.pt

Abstract

Scheduling exams and constructing a timetabling is in general a complex and difficult task. This decision problem can be approached as an optimization problem and the constraints can be categorized into two groups defined as soft and hard constraints. Finding a non overlapping exams schedule is consider an hard constraint while looking for an evenly distributed schedule and 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 and used Tabu Search to find a good solution. The tabu Search incorporates a Fuzzy Inference Ruled Based System to chose the tabu tenure of the elements in the tabu list. In addition, in each iteration the inspection of solutions in the neighborhood of a certain point is necessary and the election of an improved solution can be considered a multiple attribute decision problem. In order to rank the solutions in each neighborhood an aggregation method is proposed based on the Compromise Ratio (CR) methodology. However, we introduced a modification by considering weight functions instead of fixed weights which allows for a more flexible modeling of preferences. The chosen function should guarantee the monotonicity of the operator and we present a theoretical result regarding sufficient conditions for achieving such property.

1 Introduction

1.1 The examination timetabling problem

Many institutions and services like hospital, transportation enterprizes and educational establishments have to deal with timetabling problems. Finding a good timetabling is a crucial task necessary not only for a successful management but also to ensure the quality of the provided service. These problems have attracted considerable interest by the research community and many interesting proposals have been presented, to solve timetabling problems in sports [35],[63], transportations (bus,railways,planes) [42], [18], [43], [52], schools [5], [26], [55], [39], [58] and universities [9], [15], [17], [19], [25], [21], [22], [27], [30], [33], [36], [31], [32], [43], [45], [50], [51], [53], [59], [61], [62], [64], [67].

Although we will consider, in particular, exams timetabling the ideas presented here can be extended to many other applications, which includes not only other scheduling problems but also multicriteria problems in general. The reason to present an application to exams timetabling is justified by the affiliation of the authors and their awareness of the increased difficulty that some recent strategies have introduced in this academic task. Just as an example, we can mention the tendency towards the flexibility of curricula and the increase of the number of students enrolled in each course. The choice of a multicriteria model is based on the understating that different agents have different perspectives over the problem, and a sort of compromise is necessary. While students regard the spreading of consecutive exams as much as possible, allowing for more time of preparation, as an important feature, 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.

1.2 Contributions of this paper

In this paper we present a modification of a method used in multicriteria problems, known as Compromise Ratio [44]. Our new proposal is to replace the fixed weights by weighting functions introduced by Ribeiro and Pereira [2]. The function should be such as to ensure the monotonicity of the operator. A theoretical result is presented to establish sufficient conditions that can be used to characterize such functions. This method was used inside an automatic Tabu Search [49] and this combination of methods is also innovative and can be applied to solve different multiobjective problems. The same applies to the modified Compromise Ratio that by itself can be used as a tool in any multiattribute problem.

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 on examination timetabling, present the mathematical formulation of the problem and a brief introduction of Tabu Search. In Section 3 we present the Compromise Ratio and the main results regarding the application of weighting functions. In Section 4 we report the computational experience. Some conclusions are given in Section 5.

2 Preliminaries

2.1 Literature Review

We can find in the literature a great number of excellent contributions to exams timetabling problems (Werra 1985 [29]; Carter 1986 [23]; Carter and Laporte 1996 [24]; Burke, Jackson et al. 1997 [14]; Schaerf 1999 [58]; Petrovic and Burke 2004[58]), and very interesting surveys on the more recent approaches to this problem (Qu, Burke et al. 2006 [53]). Different methods have been suggested over the years. Regarding Clustering Methods we must mentioned the work by Desroches et al (1978 [30]) and Arani and Lotfi (1989 [7]). 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. Some of the best known methods in this category depend mainly on an ordering strategy, the most common are Saturation degree (Brelaz 1979 [11]), Largest Degree (Broder 1964 [12]), Largest Weighted Degree (Carter, Laporte et al. 1996 [20]), Largest Enrolment (Wood 1968 [65]) and Color Degree (Carter, Laporte et al. 1996 [24]). Asmuni et. al. (2004 [8]) used a Fuzzy Inference System to order the exams. In the category of the Metaheuristics it is worth mention the Tabu Search approaches of Di Gaspero and Schaerf (2001 [32]), Di Gaspero (2002 [31]) and White and Xie (2001 [64]), and applications using Simulated Annealing, such as in Thompson and Dowsland (1998 [62],1996 [61]). Still in the group of the Metaheuristics we have the Great Deluge method proposed by Dueck (1993 [34]), Burke and Newall (2003 [17]) and Burke et al (2004 [13]) and the Variable Neighbourhood Search proposed by Mladenovi and Hansen (1997 [47], 2001 [40]). In [15] Burke et al (2006) applied the VNS not as a local search but as an hyper-heuristic. Erben (2001 [36]) developed a Grouping Genetic Algorithm for the node coloring problem and consequently with applications to the exams timetabling problem. Costa e Hertz (1998 [28]) 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 (2005 [33]) followed this suggestion applying ANTCOL to timetabling problems. Memetic Search (MS) combines evolutionary algorithms with local search (Burke and Newall 1999 [16]). There are some methodologies that cannot be correctly described as a Metaheuristic such as the work of Caramia, Dell'Olmo e Italiano (2001 [19]). 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 (2007 [4]) developed an algorithm based on the Ahuja e Orlin neighborhood (2001 [6]). 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.

2.2 Mathemathical formulation of this problem

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 invigilators are fulfill. The minimization of the examination period may be a goal, but in many institutions is necessary to strictly respect a certain period and so the shortage of the exams period is not an issue. In general the requirements regarding exams timetabling can be classified as hard or soft constraints, where the first corresponds to conditions that must absolutely be verified (like those stated above) while the second represent desirable properties, considered as goals. The purpose is in general the production of a "good" calendar. Most likely, the quality of a examination calendar, from the individual point of view of each students, depends mainly on the spreading of exams, allowing for more time between consecutive exams. The criteria that allows to characterize the concept of a well distributed calendar may be difficult to model in just one objective. In fact different aspects should be consider, such as the avoidance of:

- exams in consecutive periods in the same day;
- more than one exam in the same day;
- an exam in the last period of one day and another in the first period of the next day;
- exams in consecutive days;

The degree of severity of these situations is different, but altogether they reflect what are the good properties of a timetabling. It is clear that in most situations it is not possible to verify, for all students, these four conditions as hard constraints at the risk of invalidating the construction of any calendar. So they should be set as soft constraints and addressed as objectives.
Although specifications of the problems can differ, essentially we have the following input data.

$$N = \text{number of exams.} \tag{1}$$
$$c_{ij} = \text{number of students enroled in course } i \text{ and } j \text{ for } i,j = 1,\dots,N \tag{2}$$
$$K = \text{number of courses} \tag{3}$$
$$P = \text{number of slots} \tag{4}$$
$$M = \text{number of exams inscriptions} \tag{5}$$

We encoded the solution using a vector of variables $T = (t_i), i = 1,\dots,N$, such that t_i represents the timeslot assign to exam i, and a set of variables $d_{t_i}, i = 1,\dots,N$ which represents the day corresponding to slot t_i in which exam i takes place. This last group of variables is not necessary in practice, specially if the number of slots per day is fixed, but it permits a more elegant and clear formulation.
As in [66] we have consider the following four objectives:

- The number of conflicts where students have exams in adjacent periods of the same day

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

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

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

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

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

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

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

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

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

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

2.3 Tabu search

Tabu search [37], [38] is a metaheuristic 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
- Tabu length - The length of the tabu list.
- 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 a multiobjective problem either we make an "a priori" aggregation of the objective functions or else in each iteration we calculate the value of each objective function for each point in the neighborhood and we are facing a multiattribute problem in order to elect one of those points to became the new center of the next neighborhood. We made this last option to avoid preconditionating the problem and to be able, if necessary, to change the preferences towards the objective functions along the sequence of the iterations. This could act as well as a sort of diversification strategy. The details of the application of the TS can be found in [49]. We will give here only a general overview on the main features of the TS.

2.3.1 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

2.3.2 Initial Solution

In the application of TS to the exams timetabling problem we used a graph coloring heuristic, known as "*Saturation Degree*" [11] to find a starting solution. This option was based on a paper of Carter and Laport [24] 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.

2.3.3 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 [48]. 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

For both neighborhoods the all search space was inspected in each iteration.

2.3.4 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 it is next explained.

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

The number of iterations during which a tabu remains in 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, but on the other hand it favors an intensification of the search in a region. On the 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. The importance of an automatic implementation of the TS, can be comprehended if we consider that most of the staff responsible for the elaboration of the timetabling possesses little, or most likely, no technical skills to conduct the parametrization of the TS. Ideally the algorithm should be able to automatically chose between low and high values for the tabu tenure in order to combine intensification with diversification. Since this problem can be considered a decision problem to accomplish the task of deciding for each tabu its tabu tenure we implemented a Fuzzy Rule Based System. The details of this implementation can be found in [49]. In a brief description we can add that the idea behind the FRBS is to emulate a behavior that will penalize moves, and will consequently set a high value for the tabu tenure, that in the past iterations have been often and recently been present in the tabu list. In opposition, moves rarely and not recently present in the tabu list should have lower values for the tabu tenure. The notion of the frequency in the tabu list and the last iteration where it was present in the same tabu list where recorded as *Frequency* and *Inactivity* respectively. In Figure 1 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. A 0-order Sugeno system [60] with 9 rules was implemented using as inputs linguistic variables based on *Frequency* and *Inactivity* .

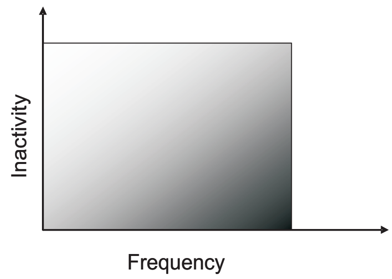

Figure 1: Tabu Tenure depending on Frequency and Inactivity

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 the 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 multiattribute decision making (MADM) problem.

3.1 Data Normalization

In multiattribute problems, in order to compare the attributes, is necessary to perform a normalization. Since the attributes may refer to different unities of measurement, to have something comparable we need to obtain values that are free of a particular scale. One common technique is the division by the maximum, but in some cases it can happen that the value of reference is not the maximum but a smaller value. It is the case when one attribute is the price but the user has a threshold that is lower than the most expensive item. In this case the utility of an item which price is above the threshold should be zero. Our approach was to make a mapping of the values on the interval $[0, 1]$ in a sort

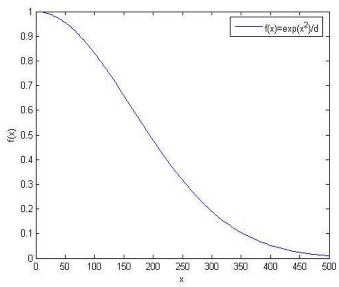

Figure 2: Gaussian mapping function

fuzzyfication approach. We used the positive tale of a gaussian function with zero mean, because of its smoothness and shape that allows a good representation of the utility of the values. In fact the 0 is transformed into the highest value of 1. It corresponds to the ideal situation of complete fulfilment of the objective function. For values near the ideal the penalization should not be to severe, but there should be a deep accentuation of this severity for values higher than an average value. We used for the four objectives $f_i(x), i = 1, \ldots, 4$ the same expression

$$\widetilde{f_i}(x) = \exp\left(\frac{-x^2}{d}\right) \text{ where } d = \frac{-m_i^2}{\log(10^{-2})} \tag{11}$$

only differing in the value of the constant m_i for $i = 1, \ldots, 4$.

$$m_1 = 0.15M \tag{12}$$
$$m_2 = 0.25M \tag{13}$$
$$m_3 = 0.10M \tag{14}$$
$$m_4 = M \tag{15}$$

where M is the total number of inscriptions in exams. These m_i values are related to a threshold that corresponds to the highest reasonable value, above which it is considered an unwilling situation. The value of 10^{-2} in (11) was set to be the minimum of $\widetilde{f_i}(x)$ for $x = m$.

3.2 Compromise Ratio

Now in each iteration of the TS, when inspecting a set of V points T_i in the neighborhood of a certain point, we evaluate these points using the four objective functions and after normalization we obtain a matrix with V objects and 4 attributes.

$$\begin{array}{ccc} & & \overbrace{\begin{array}{cccc} \widetilde{f_1} & \widetilde{f_2} & \widetilde{f_3} & \widetilde{f_4} \end{array}}^{\text{ATTRIBUTES}} \\ \text{ALTERNATIVES} & \left\{ \begin{array}{c} T_1 \\ T_2 \\ \vdots \\ T_V \end{array} \right. & \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \cdots & \cdots & \cdots & \cdots \\ x_{V1} & x_{V2} & x_{V3} & x_{V4} \end{bmatrix} \end{array}$$

To proceed for the next iteration it is necessary to chose one timetabling solution from $\{T_1, T_2, \ldots, T_V\}$ to become the next solution from which, in the next iteration, other neighborhood solutions will be searched. Most likely none of these solutions dominate the others, so it is necessary to make a choice based on the 4 criteria already mentioned. Different multiatribute methods like AHP [57], ELECTRE [56], PROMETHEE [10] could be used to produce an elected solution. Although very interesting for many applications, in this case they must be considered computationally heavy, since they have to be called as a subroutine in each iteration of the TS. Other simplest methods like MaxiMin, MiniMax and Conjuntive&Disjunctive could be implemented but they lack some sophistication that is necessary in order to produce a good strategy of choice, which is a vital step in the success of the Tabu Search. A good balance between computational simplicity and a satisfactory criteria to rank the solutions, lead to the choice of Compromise Ratio [44]. This method can be view as an extension of TOPSIS [41].

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

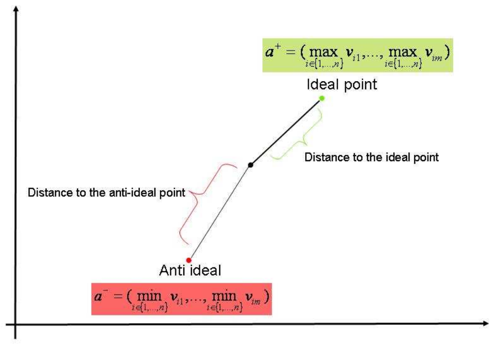

Figure 3: Distance to ideal and negative-ideal point.

the decision maker a component wise weighting of matrix $\mathbf{X} = (x_{ij})$ is performed,

$$v_{ij} = x_{ij} \times w_{ij}. \tag{16}$$

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

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m}(a_j^+ - v_{ij})^p} \quad , \forall i = 1, ..., n \tag{17}$$

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m}(v_{ij} - a_j^-)^p} \quad , \forall i = 1, ..., n. \tag{18}$$

Given these distances we can use the following parameterized ratio

$$\xi_p(T_i) = \theta \times \frac{D_{p^-}(T^+) - D_p^+(T_i)}{D_{p^-}(T^+) - D_{p^+}(T^+)} + (1-\theta) \times \frac{D_p^-(T_i) - D_{p^-}(T^-)}{D_{p^+}(T^-) - D_{p^-}(T^-)} \tag{19}$$

where

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

in order to rank the alternatives.

3.3 Weighting functions

The weights in (16) are user defined and have an highly influential effect in the final ranking of the alternatives. Its choice must be careful and since the goal was to developed an as much as possible automatic algorithm the definition of the weights by the user seemed a drawback in this method. In addition to assume that we can capture the preferences by a simple linear model seemed to reductive. Sometimes the decision maker does not react linearly and independently in regard to the attributes. 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 also to be 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. There is, the value of one attribute influences the weight given to another attribute. The same happens for the attributes regarding the exams timetabling problem. For instance, if a calendar is such that many students have more than one exam in the same day, then it is necessary to reinforce the weight that penalizes the existence of exams in consecutive periods. To model situations like these Ribeiro and Pereira [2] proposed the use of weighting functions instead of fixed weights. The gain is that we can model preferences, as well as the behavior of the decision maker, better than if we simply use constant weights.The weights are defined by the following expression.

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

where the g_j are the weight generating functions.

Mixture operators, in the context of aggregation operators where introduced in [54] and [1]. In [2], [3] we can find some interesting applications. In this work we have used weighting function in the context of the Compromise Ratio. By replacing in (16) the weights by weighting functions we developed a new procedure for ranking the alternatives that presents an higher and more realistic way of modeling preferences.

3.4 Modified Compromise Ratio

If we use weighting functions then the ideal point becomes

$$(g_1(1), \dots, g_m(1)) \tag{20}$$

and the negative-ideal is chosen as the vector

$$(0, \dots, 0)\,. \tag{21}$$

The equations analogue to (17) and (18) are

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(\frac{g_j(1) - x_{ij} g_j(x_{ij})}{\sum_{k=1}^{m} g_k(x_{ij})} \right)^p} \quad , \forall i = 1, ..., n \tag{22}$$

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(\frac{x_{ij} g_j(x_{ij})}{\sum_{k=1}^{m} g_k(x_{ij})} \right)^p} \quad , \forall i = 1, ..., n. \tag{23}$$

Now the issue is to chose the functional expression of the weighting functions. We know that the operator ξ_p is not monotonic for all weighting functions. In order to guarantee the monotonicity of ξ_p we obtained a condition similar to those presented in [54],[1] and [46].

Theorem 1. *The inequalities*

$$0 \le g_k \le 1, \qquad \forall k \in \{1, \dots, m\} \tag{24}$$

$$\frac{\partial g_k}{\partial x_{ik}} \ge 0, \qquad \forall k \in \{1, \dots, m\} \tag{25}$$

$$x_{ik}^{p-1} g_k^p - \frac{\partial g_k}{\partial x_{ik}} \ge 0, \qquad \forall k \in \{1, \dots, m\} \tag{26}$$

are sufficient conditions for the monotonicity of ξ_p as defined in (19).

Proof To guarantee the monotonicity of ξ_p it is suffices to ensure that $\frac{\partial \xi_p}{\partial x_{ik}} \geq 0$. Since

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

then if $\frac{\partial D_p^+}{\partial x_{ik}} \leq 0$ and $\frac{\partial D_p^-}{\partial x_{ik}} \geq 0$ we have $\frac{\partial \xi_p}{\partial x_{ik}} \geq 0$. Let us first prove that

$$\frac{\partial D_p^+}{\partial x_{ik}} \leq 0.$$

By definition

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(\frac{g_j(1) - x_{ij} g_j(x_{ij})}{\sum_{t=1}^{m} g_t(x_{it})} \right)^p} \quad , \text{ for } p \geq 1 \tag{27}$$

To simplify the writing let us consider the expressions

$$\gamma_q(x_i) = \left(\frac{g_q(1) - x_{iq} g_q(x_{iq})}{\varepsilon(x_i)} \right)^p \text{ for } q = 1, \ldots, m \tag{28}$$

where $x_i = (x_{i1}, \ldots, x_{im})$, $\varepsilon(x_i) = \sum_{t=1}^{m} g_t(x_{it})$ and

$$\sigma(x_i) = \sum_{q=1}^{m} \gamma_q(x_i). \tag{29}$$

Now

$$D_p^+(T_i) = \sqrt[p]{\sigma(x_i)} = \sqrt[p]{\sum_{q=1}^{m} \gamma_q(x_i)} = \sqrt[p]{\sum_{j=1}^{m} \left(\frac{g_j(1) - x_{ij} g_j(x_{ij})}{\varepsilon(x_i)} \right)^p} \quad , \text{ for } p \geq 1 \tag{30}$$

we have

$$\frac{\partial D_p^+}{\partial x_{ik}} = \frac{1}{p} \frac{\frac{\partial \sigma}{\partial x_{ik}}}{(D_p^+(T_i))^{p-1}} \tag{31}$$

Once $\frac{\partial \varepsilon}{\partial x_{ik}} = \frac{\partial g_k}{\partial x_{ik}}$ and $\frac{\partial \sigma}{\partial x_{ik}} = \sum_{q=1}^{m} \frac{\partial \gamma_q(x_i)}{\partial x_{ik}}$ then when $q = k$ we have

$$\frac{\partial \gamma_q}{\partial x_{ik}} = p \left(\frac{g_k(1) - x_{ik} g_k(x_{ik})}{\varepsilon(x_i)} \right)^{p-1} \frac{\left(-x_{ik} \frac{\partial g_k}{\partial x_{ik}} - g_k(x_{ik}) \right) \varepsilon(x_i) - \frac{\partial g_k}{\partial x_{ik}} \left(g_k(1) - x_{ik} g_k(x_{ik}) \right)}{\varepsilon(x_i)^2} \tag{32}$$

Using the left inequality of hypothesis (24) we may conclude that

$$\varepsilon(x_i) \geq 0. \tag{33}$$

Since $x_{ij} \leq 1$ and using condition (25) we know that

$$g_k(1) - x_{ik} g_k(x_{ik}) \geq 0. \tag{34}$$

From (33) and (34) and taking in consideration that $p \geq 1$ is guaranteed that

$$p \left(\frac{g_k(1) - x_{ik} g_k(x_{iq})}{\varepsilon(x_i)} \right)^{p-1} \geq 0 \tag{35}$$

Since $x_{ij} \geq 0$ and using condition (33), hypotheses (25) and the left side of condition (24) we have

$$\left(-x_{ik}\frac{\partial g_k}{\partial x_{ik}} - g_k(x_{ik})\right)\varepsilon(x_i) \leq 0. \tag{36}$$

From (34) and (25) we have

$$-\frac{\partial g_k}{\partial x_{ik}}\left(g_k(1) - x_{ik}g_k(x_{ik})\right) \leq 0. \tag{37}$$

So from (35), (36) and (37) we have

$$\frac{\partial \gamma_q}{\partial x_{ik}} \leq 0 \tag{38}$$

Now when $q \neq k$ we have

$$\frac{\partial \gamma_q}{\partial x_{ik}} = p\left(\frac{g_q(1) - x_{iq}g_q(x_{iq})}{\varepsilon(x_i)}\right)^{p-1}\frac{\frac{-\partial g_k}{\partial x_{ik}}\left(g_q(1) - x_{iq}g_q(x_{iq})\right)}{\varepsilon(x_i)^2}$$

Now by (35) and (37) we have

$$\frac{\partial \gamma_q}{\partial x_{ik}} \leq 0. \tag{39}$$

From (38) and (39) we have

$$\frac{\partial \sigma}{\partial x_{ik}} \leq 0 \tag{40}$$

Since $D_p^+ \geq 0$ and $p \geq 1$ then

$$\frac{1}{p(D_p^+(T_i))^{p-1}} \geq 0 \tag{41}$$

Through (31), (40) and (41) we may conclude that

$$\frac{\partial D_p^+}{\partial x_{ik}} \leq 0 \tag{42}$$

Now we will proof that

$$\frac{\partial D_p^-}{\partial x_{ik}} \geq 0.$$

We have

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m}\left(\frac{x_{ij}g_j(x_{ij})}{\sum_{t=1}^{m} g_t(x_{it})}\right)^p} \quad , \text{ for } p \geq 1 \tag{43}$$

Again to simplify the writing let us consider the expressions

$$\overline{\gamma}_q(x_i) = \left(\frac{x_{iq}g_q(x_{iq})}{\overline{\varepsilon}(x_i)}\right)^p \text{ for } q = 1, \ldots, m \tag{44}$$

where $x_i = (x_{i1}, \ldots, x_{im})$, $\overline{\varepsilon}(x_i) = \sum_{t=1}^{m} g_t(x_{it})$ and

$$\overline{\sigma}(x_i) = \sum_{q=1}^{m} \overline{\gamma}_q(x_i). \tag{45}$$

Now we have

$$\frac{\partial D_p^-}{\partial x_{ik}} = \frac{1}{p}\frac{\frac{\partial \overline{\sigma}}{\partial x_{ik}}}{(D_p^-(T_i))^{p-1}} \tag{46}$$

Once $\frac{\partial \overline{\varepsilon}}{\partial x_{ik}} = \frac{\partial g_k}{\partial x_{ik}}$ and $\frac{\partial \overline{\sigma}}{\partial x_{ik}} = \sum_{q=1}^{m} \frac{\partial \overline{\gamma}_q(x_i)}{\partial x_{ik}}$ then when $q = k$ we have

$$\begin{aligned}
\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} &= p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i) - \frac{\partial g_k}{\partial x_{ik}}(x_{ik}g_k(x_{ik}))}{\overline{\varepsilon}(x_i)^2} && (47)\\
&= p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} && (48)\\
&\quad -p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}(x_{ik}g_k(x_{ik}))}{\overline{\varepsilon}(x_i)^2} && (49)
\end{aligned}$$

Using the right inequality of hypothesis (24) and the fact that $x_{ik} \leq 1$ we may conclude that

$$\begin{aligned}
\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} &\geq p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} && (50)\\
&\quad -p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}(x_{ik}g_k(x_{ik}))}{\overline{\varepsilon}(x_i)^2} && (51)
\end{aligned}$$

When $q \neq k$ we have

$$\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} = p\left(\frac{x_{iq}g_q(x_{iq})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\frac{-\partial g_k}{\partial x_{ik}}(x_{iq}g_q(x_{iq}))}{\overline{\varepsilon}(x_i)^2}$$

Now since $g_k(x_{ik}) \leq 1$ and $x_{ik} \leq 1$ we have

$$\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} \geq -p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}(x_{ik}g_k(x_{ik}))}{\overline{\varepsilon}(x_i)^2} \tag{52}$$

From (51) and (52) we obtain

$$\begin{aligned}
\frac{\partial \overline{\sigma}}{\partial x_{ik}} &\geq p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} && (53)\\
&\quad -p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}}{\overline{\varepsilon}(x_i)^2}\sum_{s=1}^{m}(x_{is}g_s(x_{is})) && (54)
\end{aligned}$$

Since $x_{is} \leq 1$ we have $\overline{\varepsilon}(x_i) \geq \sum_{s=1}^{m}(x_{is}g_s(x_{is}))$ and we conclude that

$$\begin{aligned}
\frac{\partial \overline{\sigma}}{\partial x_{ik}} &\geq p\left(\frac{x_{ik}g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} && (55)\\
&\quad -p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}}{\overline{\varepsilon}(x_i)^2}\overline{\varepsilon}(x_i) && (56)
\end{aligned}$$

$$\frac{\partial \overline{\sigma}}{\partial x_{ik}} \geq p\frac{1}{\overline{\varepsilon}(x_i)^p}g_k(x_{ik})^{p-1}x_{ik}^p\frac{\partial g_k}{\partial x_{ik}} + p\frac{1}{\overline{\varepsilon}(x_i)^p}\left(x_{ik}^{p-1}g_k(x_{ik})^p - \frac{\partial g_k}{\partial x_{ik}}\right) \tag{57}$$

Given conditions (24), (25), $\overline{\varepsilon}(x_i) \geq 0$ and $p \geq 1$ we obtain

$$p\frac{1}{\overline{\varepsilon}(x_i)^p}g_k(x_{ik})^{p-1}x_{ik}^p\frac{\partial g_k}{\partial x_{ik}} \geq 0. \tag{58}$$

The hypotheses (26), $\overline{\varepsilon}(x_i) \geq 0$ and $p \geq 1$ ensure that

$$p\frac{1}{\overline{\varepsilon}(x_i)^p}\left(x_{ik}^{p-1}g_k(x_{ik})^p - \frac{\partial g_k}{\partial x_{ik}}\right) \geq 0. \tag{59}$$

With (59) we can finally prove that $\frac{\partial D_p^-}{\partial x_{ik}} \geq 0$ and conclude the proof.■

Next it was necessary to find weighting functions verifying the conditions of Theorem 1. It was easy to verify that linear functions couldn't do and so other functions were investigated.

Theorem 2. *The following functions verify the hypotheses (24) to (26) of Theorem 1.*

$$g(x) = (a-d)x^p + d \tag{60}$$

$$g(x) = d\left(\frac{a}{d}\right)^{x^p} \tag{61}$$

$$g(x) = \log\left(1 + e^{a-d} - 1)x^p\right) + d \tag{62}$$

Where a represents the importance (weight) of the attribute when its satisfaction is maximum and it value belongs to the unit interval, and d represent the importance (weight) of the attribute when its satisfaction is minimum. However d belongs to the interval $[\text{lower\_bound}, a]$. The lower_bound value is derived from the same condition referred above. In the computational experience we used for all attributes the function (60).

4 Computational experience

The algorithm for this multiple objective approach for timetabling problems was tested using the Toronto's benchmark data set [24]. The main characteristics are displayed in Table 1. The last column D.C.M. stands for Density of the Conflict Matrix. This matrix

Data set	Institution	N° of Periods	N° of exams	N° of students	D.C.M.
car-f-92	Carleton Uni., Ottawa	32	543	18419	0,14
car-s-91	Carleton Uni., Ottawa	35	682	16925	0,13
ear-f-83	Earl Haig Collegiate Inst., Toronto	24	190	1125	0,29
hec-s-92	Ecole des Hautes Et. Com., Montreal	18	81	2823	0,42
kfu-s-93	King Fahd Uni., Dharan	20	461	5349	0,06
rye-s-93	Ryeson Uni., Toronto	23	486	11483	0,07
sta-f-83	St. Andrew's Junior H.S., Toronto	13	139	611	0,14
tre-s-92	Trent Uni., Peterborough, Ontario	23	261	4360	0,18
uta-s-92	Faculty of Arts and Sciences, Uni. of Toronto	35	622	21267	0,13
ute-s-92	Faculty of Engineering, Uni. of Toronto	10	184	2750	0,08
yor-f-83	York Mills Collegiate Ins., Toronto	21	181	941	0,27

Table 1: Data set

have a number of columns and rows equal to the number of exams, and each entry (i, j) represents the number of students enrolled in both courses indexed by i and j. The percentage of nonzero elements represents its density. The higher is the value of the density the most difficult is the problem.

The experiments were performed on a Pentium Intel Core2 Duo T9400 with 2.53GHz and 3 Gb of memory. The stopping conditions used in Tabu Search were 1 hour or 25000 iterations. In all experiments the algorithm stopped after 1 hour, never reaching 25000 iterations.

The main goal was to minimize all the objectives and we also wanted to assess how the algorithm performs when using different weighting factors - a (see Equation (60)). Further, we fixed the d weighting factor in the same equation equal to the lower_bound. The computational results are depicted in the following tables. The a-weights for the different objectives are depicted on the first column of each Table. For instance, in Table 2 for data reffering to car-f92, the weight regarding first, second, third and fourth objective is 1, 1, 1, 1, respectively. We can observe that in most cases, higher values for a in one of the objectives induces a lower values for the objective value at the expenses of all the other objectives. All other objectives have worse values in favor of the objective with the highest weight.

The preliminary results obtained for the kfu-s-93, rye-s-93 and uta-s-92 data sets presented a very poor discrimination of the objectives values for the different weights. We decided to increase the stopping time from 1 hour to 2 for the previously mentioned data sets. With this change we observed an improvement on the quality of the results, as it can be seen in Tables 2 and 3, for data corresponding to kfu-s-93, rye-s-93 and uta-s-92. The same problem can be observed on car-f-92, car-s-91. The solutions seem to be very close to each other for different weight values. The algorithm seems to get trapped in a local optima regarding the data sta-f-83 because it presents the exactly same results for different weight values.

5 Conclusions

In this paper we present a method to solve a multiobjective approach to an exams timetabling problem. Timetabling problems are important and appear in many different contexts. The use of the Tabu Search is well justified due to the complexity of the problem. In each iteration of this method, in order to evaluate the neighboring solutions we used a modified version of the multiattribute method Compromise Ratio. We have proposed the replacement of the fixed weights by weighting generating functions. This new feature is more powerful and realistic in modeling the preferences of the decision maker. To guarantee the monotonicity of aggregation operator we established some sufficient conditions for the weighting generation function. These condition enable the characterization of a set of functions. The proposal of this paper can be easily adapted to other problems and the theoretical results presented for the Modified Compromise Ratio method can be useful in the context of any multiattribute problem. The computational results were satisfactory but could not be compared with other approaches in the literature.

References

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

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

[3] Marques Pereira R. A. and Ribeiro R. A. Generalized mixture operators using weighting functions : a comparative study with wa and owa. *European Journal of Operational Research*, 145:329–342, 2003.

[4] S. Abdullah, S. Ahmadi, E.K. Burke, and M. Dror. Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling. *OR Spectrum*, 29(2):351–372, 2007.

[5] D. Abramson, M. Krishnamoorthy, and H. Dang. Simulated annealing cooling schedules for the school timetabling problem. *Asia Pacific Journal Of Operational Research*, 16:1–22, 1999.

[6] R.K. Ahuja, J.B. Orlin, and D. Sharma. Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. *Mathematical Programming*, 91(1):71–97, 2001.

[7] T. Arani and V. Lotfi. A three phased approach to final exam scheduling. *IIE Transactions*, 21(1):86–96, 1989.

[8] H. Asmuni, E.K. Burke, J.M. Garibaldi, and B. McCollum. Fuzzy Multiple Ordering Criteria for Examination Timetabling. In *PATAT*, volume 3616, 2004.

[9] R. Awad and J. Chinneck. Proctor Assignment at Carleton University. *Interfaces*, 28(2):58–71, 1998.

[10] J. P. Brans and P. Vincke. A preference ranking organization method. *Management Science*, 7:647–656, 1985.

[11] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[12] S. Broder. Final examination scheduling. *Communication of the ACM*, 7:494–498, 1964.

[13] E. Burke, Y. Bykov, J. Newall, and S. Petrovic. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36(6):509–528, 2004.

[14] E. Burke, K. Jackson, JH Kingston, and R. Weare. Automated University Timetabling: The State of the Art. *The Computer Journal*, 40(9):565–571, 1997.

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

[16] EK Burke and JP Newall. A multistage evolutionary algorithm for the timetable problem. *Evolutionary Computation, IEEE Transactions on*, 3(1):63–74, 1999.

[17] EK Burke and JP Newall. Enhancing Timetable Solutions with Local Search Methods. *Lecture Notes In Computer Science*, pages 195–206, 2003.

[18] A. Caprara, M. Fischetti, PL Guida, M. Monaci, G. Sacco, and P. Toth. Solution of real-world train timetabling problems. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 10, 2001.

[19] M. Caramia, P. Dell Olmo, and G.F. Italiano. New Algorithms for Examination Timetabling. *Lecture Notes In Computer Science*, pages 230–242, 2001.

[20] M. W. Carter and G. Laporte. Recent Developments in Practical Examination Timetabling. In *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling*, 1996.

[21] M. W. Carter, G. Laporte, and J.W. Chinneck. A general examination scheduling system. *Interfaces*, 24(3):109–120, 1994.

[22] Michael W. Carter and Gilbert Laporte. Recent developments in practical examination timetabling. In *Selected papers from the First International Conference on Practice and Theory of Automated Timetabling*, pages 3–21, London, UK, 1996. Springer-Verlag.

[23] M.W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2):193–202, 1986.

[24] M.W. Carter, G. Laporte, and S.Y. Lee. Examination timetabling: Algorithmic strategies and applications. *OR Journal(Journal of the Operational Research Society)*, 47(3):373–383, 1996.

[25] S. Casey and J. Thompson. GRASPing the Examination Scheduling Problem. *Lecture Notes In Computer Science*, pages 232–246, 2003.

[26] A. Colorni, M. Dorigo, and V. Maniezzo. Metaheuristics for High School Timetabling. *Computational Optimization and Applications*, 9(3):275–298, 1998.

[27] PH Corr, B. McCollum, MAJ McGreevy, and P. McMullan. A New Neural Network Based Construction Heuristic for the Examination Timetabling Problem. *Lecture Notes In Computer Science*, 4193:392, 2006.

[28] D. Costa and A. Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1998.

[29] D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985.

[30] S. Desroches, G. Laporte, and JM Rousseau. Horex: A computer program for the construction of examination schedules. *INFOR*, 16:294–298, 1978.

[31] L. Di Gaspero. Recolour, shake and kick: A recipe for the examination timetabling problem. In *Proceedings of the Fourth International Conference on the Practice and Theory of Automated Timetabling, Gent, Belgium*, pages 404–407, 2002.

[32] L. Di Gaspero and A. Schaerf. Tabu Search Techniques for Examination Timetabling. *Lecture Notes In Computer Science*, pages 104–117, 2001.

[33] KA Dowsland and JM Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438, 2005.

[34] G. Dueck. New optimization heuristics: The great deluge algortihm and the record-to-travel. *Journal of Computational Physics*, 104:86–92, 1993.

[35] K. Easton, G. Nemhauser, and M. Trick. Sports scheduling. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

[36] W. Erben. A Grouping Genetic Algorithm for Graph Colouring and Exam Timetabling. *Lecture Notes In Computer Science*, pages 132–158, 2001.

[37] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[38] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publisher, 1997.

[39] M.P. Hansen and R.V.V. Vidal. Planning of high school examinations in Denmark. *European Journal of Operational Research*, 87(3):519–534, 1995.

[40] P. Hansen and N. Mladenovic. Variable neighbourhood search. *European Journal of Operational Research*, 130:449–467, 2001.

[41] C. L. Hwang and K. Yoon. *Multiple Attributes Decision Making Methods and Applications*. Berlin, Springer, 1981.

[42] MT Isaai and MG Singh. Hybrid applications of constraint satisfaction and metaheuristicsto railway timetabling: a comparative study. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 31(1):87–95, 2001.

[43] G. Kendall and N. Mohd Hussin. An Investigation of a Tabu Search Based Hyperheuristic for Examination Timetabling. *Selected Papers from MISTA*, pages 309–328, 2003.

[44] Deng Li. Compromise ratio method for fuzzy multi-attribute group decision making. *Applied Soft Computing*, 7:43 – 58, 2007.

[45] L.T.G. Merlot, N. Boland, B.D. Hughes, and P.J. Stuckey. A Hybrid Algorithm for the Examination Timetabling Problem. *Lecture Notes In Computer Science*, pages 207–231, 2003.

[46] Spirkova J. Mesiar R. and Vavrikova L. Weighted aggregation operators based on minimization. *Information Sciences*, 2008.

[47] N. Mladenovic and P. Hansen. Variable neighbourhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

[48] C. Morgenstern. *Algorithms for general graph coloring.* PhD thesis, Department of Computer Science, University of NewMexico, 1989.

[49] T. Pais and P. Amaral. Tabu Search with a Fuzzy Inference System : an application to examination timetabling. In *Proceedings of the PATAT08 International Conference (PATAT 2008)*, 2008.

[50] L.F. Paquete and C.M. Fonseca. A Study of Examination Timetabling with Multi-objective Evolutionary Algorithms. In *Proceedings of the 4th Metaheuristics International Conference (MIC2001)*, pages 149–153, 2001.

[51] S. Petrovic and Y. Bykov. A Multiobjective Optimisation Technique for Exam Timetabling Based on Trajectories. *Lecture Notes In Computer Science*, pages 181–194, 2003.

[52] X. Qi, J. Yang, and G. Yu. Scheduling problems in the airline industry. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

[53] R. Qu, EK Burke, B. McCollum, L.T.G. Merlot, and S.Y. Lee. A Survey of Search Methodologies and Automated Approaches for Examination Timetabling. *Computer Science Technical Report No. NOTTCS-TR-2006-4*, 2006.

[54] Marques Pereira R. and Pasi G. A. On non-monotonic aggregation: mixture operators. In *Conference on Soft and Intelligent Computing SIC, Technical University of Budapest,Hungary:*, 1999.

[55] G. Ribeiro Filho and L.A.N. Lorena. A Constructive Evolutionary Approach to School Timetabling. *Applications of Evolutionary Computing - Lecture Notes in Computer Science*, 2037:130–139, 2001.

[56] B. Roy. Classement et choix en présence de points de vue multiples (la méthode electre). *Revue d'Informatique et de Recherche Opérationelle*, 8:57–75, 1968.

[57] T. L. Saaty. *The Analytical Hierarchy Process.* New York, McGraw Hill., 1980.

[58] A. Schaerf. Local search techniques for large high school timetabling problems. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 29(4):368–377, 1999.

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

[60] M. Sugeno and GT Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.

[61] J.M. Thompson and K.A. Dowsland. Variants of simulated annealing for the examination timetabling problem. *Annals of Operations Research*, 63(1):105–128, 1996.

[62] J.M. Thompson and K.A. Dowsland. A robust simulated annealing based examination timetabling system. *Computers and Operations Research*, 25(7-8):637–648, 1998.

[63] M.A. Trick. A Schedule-Then-Break Approach to Sports Timetabling. *Lecture Notes In Computer Science*, pages 242–253, 2001.

[64] G.M. White and B.S. Xie. Examination Timetables and Tabu Search With Longer-Term Memory. *Lecture Notes In Computer Science*, pages 85–103, 2001.

[65] D.C. Wood. A system for computing examination timetables. *The Computer Journal*, 11(1):41–47, 1968.

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

[67] Y. Yang and S. Petrovic. A Novel Similarity Measure for Heuristic Selection in Examination Timetabling. *Lecture Notes In Computer Science*, 3616:247, 2005.

A Appendix - Computational Results

Table 2: Computational results

car-f-92	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	7811	5220	6566	233
(1,0.2,0.2,0.2)	7296	4262	5896	373
(0.2,1,0.2,0.2)	8074	4093	5734	330
(0.2,0.2,1,0.2)	6753	4470	6378	339
(0.2,0.2,0.2,1)	8332	4191	5626	253
car-s-91	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	11457	4010	5297	286
(1,0.2,0.2,0.2)	9764	3998	5637	343
(0.2,1,0.2,0.2)	10595	3929	5090	327
(0.2,0.2,1,0.2)	10595	3929	5090	327
(0.2,0.2,0.2,1)	11452	4015	5304	286
ear-f-83	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	2970	2379	3196	116
(1,0.2,0.2,0.2)	2576	2051	3480	424
(0.2,1,0.2,0.2)	5911	475	1602	542
(0.2,0.2,1,0.2)	5896	650	1292	367
(0.2,0.2,0.2,1)	5799	988	1535	126
hec-s-92	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	5103	475	870	170
(1,0.2,0.2,0.2)	2019	1112	3054	173
(0.2,1,0.2,0.2)	5512	250	1427	581
(0.2,0.2,1,0.2)	5385	599	800	421
(0.2,0.2,0.2,1)	5830	915	1507	84
kfu-s-93	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	9687	6230	10075	424
(1,0.2,0.2,0.2)	5123	3424	7687	481
(0.2,1,0.2,0.2)	13066	831	1307	1307
(0.2,0.2,1,0.2)	12438	1452	3359	872
(0.2,0.2,0.2,1)	14415	2198	3619	209

Table 3: Computational results

rye-s-93	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	5599	10108	15023	221
(1,0.2,0.2,0.2)	8883	9147	14007	783
(0.2,1,0.2,0.2)	22614	1659	6331	2260
(0.2,0.2,1,0.2)	21806	1686	3318	1827
(0.2,0.2,0.2,1)	25546	3936	6527	248
sta-f-83	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	8432	3195	4669	1121
(1,0.2,0.2,0.2)	8422	3202	4678	1148
(0.2,1,0.2,0.2)	8422	3202	4678	1148
(0.2,0.2,1,0.2)	8422	3202	4678	1148
(0.2,0.2,0.2,1)	9577	3254	4427	838
tre-s-92	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	4762	576	966	203
(1,0.2,0.2,0.2)	3175	978	1893	401
(0.2,1,0.2,0.2)	4920	332	1297	599
(0.2,0.2,1,0.2)	5001	554	899	356
(0.2,0.2,0.2,1)	5637	919	1471	149
uta-s-92	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	11310	1861	3188	936
(1,0.2,0.2,0.2)	8694	4962	7505	759
(0.2,1,0.2,0.2)	13767	1619	3712	2167
(0.2,0.2,1,0.2)	12505	1674	2554	1076
(0.2,0.2,0.2,1)	10606	5189	7295	600
ute-s-92	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	2384	4885	7249	72
(1,0.2,0.2,0.2)	1747	5145	7431	471
(0.2,1,0.2,0.2)	3278	3794	6628	328
(0.2,0.2,1,0.2)	3278	4126	6624	328
(0.2,0.2,0.2,1)	5427	3982	6021	34
yor-f-83	Objective 1	Objective 2	Objective 3	Objective 4
(1,1,1,1)	4346	627	1279	182
(1,0.2,0.2,0.2)	2660	1141	2076	423
(0.2,1,0.2,0.2)	4406	342	1422	607
(0.2,0.2,1,0.2)	4687	619	1137	346
(0.2,0.2,0.2,1)	4947	934	1416	99