Compromise Ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling ☆

Tiago Cardal Pais[a], Paula Amaral[b]

[a] *School of Computer Science, University of Nottingham, Nottingham UK*
[b] *Department of Mathematics and CMA, Universidade Nova de Lisboa, Portugal*

Abstract

University examination scheduling is a difficult and heavily administrative task, particularly when the number of students and courses is high. Changes in educational paradigms, an increase in the number of students, the aggregation of schools, more flexible curricula, among others, are responsible for an increase in the difficulty of the problem. As a consequence, there is a continuous demand for new and more efficient approaches. Optimisation and Constraint Programming communities have devoted considerable attention to this difficult problem. Just the definition of a satisfactory, not to mention optimal, timetabling may be complex. In fact, to characterise a timetabling solution, a single criteria may not be enough, since what may be considered good for one group of students may be regarded inappropriate for other students, or teachers. In this paper, four criteria were used to characterise the spreading of the exams over the examination period. A set of constraints regarding the non-overlapping of exams with students in common was considered. A multi-objective optimisation program was used to handle the four criteria and a Tabu Search was implemented to find a good feasible solution for this problem. Two new features to increase the automation of the algorithm were proposed. First, it uses a Fuzzy Inference Ruled Based System to choose the tabu tenure of the elements in the tabu list. Secondly, a modified version of the Compromise Ratio (CR) is proposed, where the usual fixed weights are replaced by weighting functions to rank the neighbourhood solutions in each iteration. Sufficient conditions which guarantee the monotonicity of the weighting functions are presented.

Keywords: Multi-objective examination timetabling, Compromise Ratio, Tabu Search, weighting functions.

1. Introduction

Timetabling problems are an important field in management. They have applications in many institutions and services, such as hospitals, transportation enterprises and educational establishments. Finding a good timetable is crucial, not only for a successful management, but also to ensure the quality of the service. These problems have attracted the interest of the operations research community and many contributions were made to solve timetabling problems in different areas, like in Sports ([37],[75]); Transportation ([46],[22], [64]) and Schools ([3], [28], [39], [43],[69], [72]). Universities, as well as other educational establishments, have to deal mainly with course ([70], [21], [8], [57], [2],[9],[47], [51],[49], [76], [10], [63]) and examination timetabling ([7], [17], [19], [16], [23], [27], [26], [24], [29], [32], [35], [38], [33], [34],[42], [48], [53],[54], [55], [61], [62], [65], [73], [74], [77], [79]). This paper deals only with examination timetabling, although the methodologies could be adapted to other cases and other problems.
Since the early introductory paper of Werra (1985) [31], a number of excellent contributions can

☆This paper has been supported by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through PEst-OE/MAT/UI0297/2011 (CMA).

Email addresses: txp@cs.nott.ac.uk (Tiago Cardal Pais), paca@campus.fct.unl.pt (Paula Amaral)

be found in the examination timetabling literature. Some classical references are: Carter (1986) [25], Carter and Laporte (1996) [24], Burke, Jackson et al (1997) [15], Schaerf (1999) [69]. The more recent survey of Qu et al (2009) [65] is also worth mentioning. Different methods have been proposed over the years. For example, Clustering Methods were described in Desroches et al (1978) [32] and Arani and Lotfi (1989) [5]. The examination timetabling problem can be modelled as a node colouring problem in a graph. For this reason, most of the heuristics for node colouring can also 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, Brélaz 1979 [12]; Largest Degree, Broder 1964 [13]; Largest Weighted Degree, Carter, Laporte et al. 1996 [24]; Largest Enrolment, Wood 1968 [78] and Colour Degree, Carter, Laporte et al. 1996 [24]. A Fuzzy Inference System to order the exams is proposed by Asmuni et. al. (2004) [6]. In the category of Meta-heuristics, there are some successful Tabu Search approaches (Di Gaspero and Schaerf (2001) [34], Di Gaspero (2002) [33] and White and Xie (2001) [77]) and applications of Simulated Annealing (Thompson and Dowsland (1998) [74], (1996) [73]). Other noteworthy meta-heuristics are the Great Deluge Method, Dueck (1993) [36], Burke and Newall (2003) [17] and Burke et al (2004) [14], and the Variable Neighbourhood Search (VNS), Mladenovic and Hansen (1997) [58], (2003) [44]. In Burke et al (2006) [16], the authors applied the VNS not as a local search but as a hyper-heuristic. Erben (2001) [38] developed a Grouping Genetic Algorithm for the node colouring, which also applies to the examination timetabling problem. Costa and Hertz (1998) [30] developed a method based on Ant Colony (ANTCOL) for the node colouring problem and suggested the application to timetabling problems, as accomplished by Dowsland and Thompson (2005) [35]. Memetic Search (MS) combines evolutionary algorithms with local search. Its use for the examination timetabling problem was proposed by Burke and Newall (1999) [18]. There are some methodologies that cannot be correctly described as meta-heuristics, such as the work of Caramia, Dell'Olmo and Italiano (2001) [23]. The 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) [1] developed an algorithm based on the Ahuja and Orlin neighbourhood, Ahuja et al (2001) [4]. Even though this last method is computationally heavy, it presents some of the best results for a collection of instances frequently used by many researchers in examination timetabling problems.

In recent years, the tendency towards the flexibility of curricula and the increase in the number of students enrolled in each course have increased the difficulty of examination scheduling. In addition, the different agents involved in this problem have different perspectives and a compromise is necessary. In fact, students regard the spreading of consecutive exams as an important feature, since it allows more time for preparation, while in general teachers tend to favour a shorter examination period length to have more time to prepare the next semester and to do research. To model these preferences, a multi-objective approach was used. To find a good solution for this hard optimisation problem, a Tabu Search (TS) was implemented. Two particular features for TS were proposed and are described in the next subsection.

1.1. Contributions of this paper

One important aspect of methodologies for examination timetabling is automation. In fact, the task of scheduling examinations is performed by staff members whose expertise in computational and optimisation methods may not be assumed. Methods that depend on parameter-tuning or on experts' decisions are impractical in this case. Automation is the main motivation for the work presented in this paper. Tabu Search has proved to be an efficient approach for examination scheduling, but its drawback is the tuning of the tabu tenure, which has a great influence in the performance of the algorithm and is also a cumbersome task. To overcome this problem, an automatic tuning strategy for setting the tabu tenure was implemented which was adapted from [60] to the multi-objective case.

In every iteration of the TS, a base solution was used to define a neighbourhood from which a new solution is chosen as the starting point for the next iteration. Each solution has a set of attributes which correspond to the multiple objectives. The evaluation and ranking of the solutions in the neighbourhood is a multi-criteria decision problem. Compromise Ratio [50] is a known method in multi-criteria for ranking points with a set of attributes. In this paper, a modification of the Compromise Ratio is proposed which consists in the replacement of fixed weights by weighting functions [66]. The advantage of using weighting functions is a more flexible modelling of preferences. Once these are defined, they capture the essence of the decision-maker reasoning, henceforth, avoiding continuous human interference and ensuring algorithm automation. The idea of introducing weighting functions in Compromise Ratio is one of the major contributions of this paper. The weighting functions must ensure the monotonicity of the operator. In order to characterise such function, a set of sufficient conditions were established. The theoretical results with sufficient conditions for the monotonicity of this operator presented in Theorem 1 and the identification of a set of functions that verify those conditions described in Theorem 2 can also be considered important contributions to this paper. The merits of this approach are clearly not only conceptual but also practical. Although this technique was applied to examination timetabling, it may be extended to other applications. In particular, the modifications introduced in the Compromise Ratio can be applied to general multi-attribute problems.

1.2. Structure of the paper

After the literature review and motivation, this paper is structured in the following way. Section 2 describes the mathematical formulation of the problem. The new proposed features related to the implementation of the Tabu Search are presented in Section 3. The management of the tabu tenure using a Fuzzy Rule Based System (FRBS) is studied in Section 3.5. The multi-criteria problem related to the neighbourhood evaluation is explained in Section 4. In this same Section, the Compromise Ratio method and the main results regarding the application of weighting functions are also addressed. In Section 5, the computational results are reported. Some conclusions are given in Section 6.

2. Model assumptions and mathematical formulation

A general examination timetabling problem consists of finding such a schedule where the number of overlapping exams for each student is minimal or none. Moreover, other requirements, such as rooms and invigilators, also need to be fulfilled. Usually, the most difficult part of the problem is to find a schedule that has a good distribution of the exams over the examination period. In some cases, this period is not previously fixed and the minimisation of the examination period length may be a goal. In this case, a set of constraints are defined to guarantee that a schedule does not have any overlapping exams with students in common. However, in most cases, the examination period is previously established, since the academic calendar must be defined at the beginning of the academic year, prior to students' enrolment in courses. Generally, past experience is sufficient to dictate a comfortable choice of the examination period length. Regarding rooms and invigilators, many authors prefer to deal with these issues in a second phase, since these constraints are less restrictive and easier to verify. Following these arguments, we will consider in this paper that:

- the period of exams is predefined and corresponds to a sequence of time slots;
- the overlapping of exams with candidates in common is not allowed;
- invigilators and rooms are not considered;
- the goal is to have a good distribution of the exams over the evaluation period.

The next question is how to define these goals as an objective function. The quality of an examination schedule from the individual point of view of each student will most likely depend on the spreading of exams, which allows for more preparation time between consecutive exams. Let,

$$N = \text{number of exams.} \quad (1)$$

$$c_{ij} = \text{number of students enrolled in course } i \text{ and } j \text{ for } i, j = 1, \ldots, N. \quad (2)$$

$$P = \text{number of slots.} \quad (3)$$

The slots assigned to pairs of exams i and j where $c_{ij} \neq 0$ should be as distant as possible. The total number of common enrolments, $M = \sum_{i=1}^{N-1} \sum_{j>i}^{N} c_{ij}$, gives a good measure of the difficulty in achieving this goal. The characterisation of a well distributed calendar is also difficult to model in just one objective. In fact, different aspects should be considered, such as the avoidance for every pair of exams i and j such that $c_{ij} \neq 0$, of their schedule:

1 - in consecutive periods on the same day;

2 - on the same day;

3 - in overnight consecutive periods (except from Saturday to Monday);

4 - on consecutive days.

The order by which the above criteria are presented is related to their degree of undesirability. For instance, it is worse to have two exams in consecutive slots on the same day than in consecutive days. The first criterion is only applied if there is more than one slot per day. The second criterion only makes sense if there are more than two daily periods, for instance, one period in the morning and two in the afternoon. It is clear that in most situations it is not possible to enforce all four conditions for all students as hard constraints. This would make the search for a feasible solution almost impracticable. Hence, they should be set as soft constraints and addressed as goals. The minimisation of these four criteria will produce desirable timetables.
To encode the solution, a vector of variables $T = (t_i), i = 1, \ldots, N$ was considered, where t_i represents the time slot assign to exam i. The set of variables $d_{t_i}, i = 1, \ldots, N$, represent the day associated to timeslot slot t_i in which exam i takes place. Their definition is convenient for a more elegant and clearer formulation. However, this definition is not necessary in practice, especially if the number of slots per day is fixed. When for i and $j \in \{1, \ldots, N\}, i \neq j$, $c_{ij} \neq 0$, then c_{ij} students are affected by the scheduled vicinity of exams i and j. In the following, this situations are described as producing c_{ij} "conflicts" for simplicity. As in [20] the following four objectives are considered:

- Total number of conflicts generated by the occurrence of exams in adjacent slots on the same day,

$$f_1(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.adjs(t_i, t_j) \quad , \text{where } adjs(t_i, t_j) = \begin{cases} 1 & \text{if } (|t_i - t_j| = 1) \wedge (d_{t_i} = d_{t_j}) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- Total number of conflicts generated by the occurrence of two or more exams on the same day,

$$f_2(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.sday(t_i, t_j) \quad , \text{where } sday(t_i, t_j) = \begin{cases} 1 & \text{if } d_{t_i} = d_{t_j} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

- Total number of conflicts generated by the occurrence of exams in overnight adjacent periods,

$$f_3(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.ovnt(t_i, t_j) \quad , \text{where } ovnt(t_i, t_j) = \begin{cases} 1 & \text{if } (|t_i - t_j| = 1) \wedge (|d_{t_i} - d_{t_j}| = 1) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

- Total number of conflicts generated by the occurrence of exams in adjacent days,

$$f_4(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij}.adjd(t_i, t_j) \quad , \text{where } adjd(t_i, t_j) = \begin{cases} 1 & \text{if } |d_{t_i} - d_{t_j}| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

From (4) to (7) it follows that $f_1(T) \leq f_2(T)$ and $f_3(T) \leq f_4(T)$. A single set of hard constraints was considered, to guarantee that no student has more than one exam in the same slot,

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

3. Tabu search

Tabu search [40], [41] is a meta-heuristic that has successfully been applied to find good feasible solutions for hard optimisation problems. In general, it can be described as an iterative neighbourhood search method which incorporates techniques for escaping local optima while trying to avoid cycling through the construction of a tabu list. The tabu list consists of information that tries to prevent revisiting points. Each element in the list is a partial or a complete solution, or just a simple rule. An initial feasible solution (vector of variables) is necessary. A general and basic TS iteration k, consists in generating from a known feasible solution T_k points in its neighbourhood that are not tabu. A neighbourhood of a solution T_k consists of a collection of solutions (unequivocally obtained from T_k by the application of a set of operations) that differ from T_k in the value assigned to a group of variables. In general, these points are evaluated using an objective function and the one that has the best evaluation is chosen as the new search point, T_{k+1}, as the starting point for the next iteration. To avoid revisiting this same point, the tabu list is updated. The search point, or an action related to it (a move), is added to the tabu list and remains there for a number of iterations which are designated as "tabu tenure". However, the tabu status can be revoked by an aspiration criterion. This process is repeated until some termination criteria are reached, for instance the maximum number of iterations. There are many interesting additional refinements that can greatly increase the performance of the TS. In summary, a first level Tabu Search (TS) uses the following concepts in each k iteration.

- Current (search) solution T_k.
- Search Neighbourhood - Points that will be inspected from the current solution.
- Move - A set of simple operations on T_k that generates a particular neighbour solution.
- Evaluation - A procedure to evaluate the points in the neighbourhood.
- Tabu list - The tabu moves that are not allowed in the current iteration.
- Tabu tenure - The duration (number of consecutive iterations) of the tabu status.
- Tabu length - The length of the tabu list.
- Aspiration criteria - Enables to override the tabu status.

In a multi-objective problem either "a priori" aggregation of the objective functions is performed or, in each iteration, the value of each objective function is calculated for each point in the neighbourhood leading to a multi-attribute problem. In this paper this last option was implemented to avoiding preconditioning, and to be able to change the preferences towards the different criteria along the sequence of iterations. This could also act as a sort of diversification strategy.

The details of the application of the TS can be found in [60], while a general overview of the main features of the TS is presented next.

3.1. Solution Encoding

As mentioned before, the solution is encoded by a vector with a dimension equal to the number of exams. The integer value recorded in the i-component of the vector is t_i and corresponds to the time slot assigned to exam i. A small example is given in Table 1, where exams $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ are assigned to slot 1 and slot 2, respectively.

Table 1: A small example solution T_0 for an exam timetabling problem with 8 exam and 2 time slots

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

3.2. Initial Solution

A graph colouring heuristic known as "*Saturation Degree*" [12], based on the work of Carter and Laporte [24], was used to find a starting solution. This heuristic has a good performance in comparison with several other heuristic approaches. This greedy heuristic mainly consists in ranking exams by increasing order of the number of their still available slots. For instance, if exam i and j can only be assigned to two and three slots, respectively, then exam i is ranked before exam j. After this ranking is established, the first exam is assigned to the first available slot and the ranking of exams is updated. In case of a tie, the preference is given to the exam with more students.

3.3. Neighbourhood

A feasible neighbourhood of a solution T_0 consists of a set of feasible solutions that can be obtained from T_0 by the application of a rule that changes the value of a subset of variables in T_0. In our case, it consisted of different assignments of slots to some exams and two different neighbourhoods were considered. An elementary one, corresponding to a given timetable T_0, to all timetabling T_i differing from T_0 in the assignment of just one exam. For this neighbourhood, a move consists of a slot change for a given exam. Considering a pair (i, j) where i, j are the exam and slot number, respectively, a move is represented by the change of j to k.

$$(i, \mathbf{j}) \rightarrow (i, \mathbf{k})$$

For example, considering again the set of 8 exams and 2 slots, a possible neighbour of T_0 is timetable T_i where exam 3 changes from slot (period) 1 to 2.

T_0

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

$$(3, \mathbf{1}) \rightarrow (3, \mathbf{2})$$

T_i

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

The second neighbourhood is based on *Kempe chains* as introduced by Morgenstern [59]. A neighbourhood of timetable T_0 is a set of feasible timetables differing from T_0 by the exchange of exams between two time slots. A move corresponds to a feasible swapping of a subset of exams between two periods.

$$\begin{array}{lcl} (i, \mathbf{j}) & \rightarrow & (i, \mathbf{s}), \text{ for } i \in I \\ (r, \mathbf{s}) & \rightarrow & (r, \mathbf{j}), \text{ for } r \in R. \end{array}$$

The neighbourhood based on *Kempe chains* is initiated with a single move,

$$(i, \mathbf{j}) \rightarrow (i, \mathbf{s}) \text{ for } i \in I_0 = \{i_0\} \subseteq J,$$

where J and S are the sets of exams previously assigned to slot $\mathbf{j}$ and $\mathbf{s}$, respectively. When exam i_0 is moved to the new time slot $\mathbf{s}$, a chain of simple movements is triggered. All scheduled exams in period $\mathbf{s}$ that have a conflict with i_0 ($r \in R_0 \subseteq S$) are moved to slot $\mathbf{j}$. If this set of exams R_0 has any conflict in slot $\mathbf{j}$, the previously scheduled exams in $\mathbf{j}$, in conflict with R_0, say $I_1 \subseteq J$, are moved to period $\mathbf{s}$. This chain of simple movements continues until there are no conflicts between exams. The smallest *Kempe chain* move is a swap of periods by one exam. In order to preserve feasibility, it may happen that these exchanges must be continued between the exams of the two slots as described above. On the other hand, a complete transfer of exams between two slots may

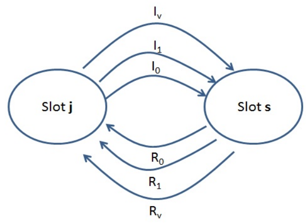

Figure 1: Sequence of moves

happen. It is noteworthy that a feasible solution will always be reached given any two slots. For both neighbourhoods, a complete inspection of the neighbouring solutions was performed.

For instance, given the example above of timetable T_0, a neighbouring solution T_i is obtained when exams 1 and 2 in period 1 and exams 5,6 and 7 in period 2 swap periods.

T_0	Exams	**1**	**2**	3	4	**5**	**6**	**7**	8
	Slot	**1**	**1**	1	1	**2**	**2**	**2**	2
	$(i, \mathbf{1})$	$\longrightarrow$	$(i, \mathbf{2},)$ for $i \in \{1, 2\}$						
	$(r, \mathbf{2})$	$\longrightarrow$	$(r, \mathbf{1})$, for $r \in \{5, 6, 7\}$						
T_i	Exams	**1**	**2**	3	4	**5**	**6**	**7**	8
	Slot	**2**	**2**	1	1	**1**	**1**	**1**	2

3.4. Memory

The memory management depends on the considered neighbourhood. For the simple neighbourhood, the index of the exam that is moved is recorded. As a consequence, in a number of iterations equal to the tabu tenure, the time slot of this exam is not allowed to change. For the *Kempe chains* neighbourhood, recording all the chain of moves required too much time and memory. On the other hand, recording all the exams that changed period may lead to an over restricting tabu list. After a few iterations, it could happen that all possible movements are tabu. As a compromise, a pair consisting of both the exam and the time slot of each exam involved in the chain of movements was recorded. So if exam i moved to slot k then the pair (i, k) was added to the tabu list. Exam i is allowed to change from time slot k to r, but cannot return to time slot k until (i, k) remains in the tabu list. In each iteration, the tabu tenure of each element in the tabu list is decreased by one unit. The length of the tabu status for each move is determined individually using a FRBS, as in [60], but adapted to the multi-objective case as it is next described.

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 a few iterations a local optima is revisited and so the algorithm goes into a loop. On the other hand, it favours an intensification of the search in a region. If the tabu tenure is high, the search space is diversified, but a refined local search is less likely to occur and good solutions may go unnoticed. In practice, it is common to run the method repeatedly for the same instance while varying the value of the tabu tenure. This task is cumbersome if conducted manually. The importance of an automatic implementation of the TS is paramount if we consider that most of the staff responsible for the creation of the timetables has no technical skills to conduct the parametrisation of the TS. Ideally, the algorithm should be able to automatically choose between low and high values for the tabu tenure in order to combine intensification with diversification. This tuning strategy can be formulated as a decision problem. To perform the task of deciding the length of the corresponding tabu tenure, a Fuzzy Rule Based System (FRBS) was implemented in each iteration and for each tabu item. The details of this implementation can be found in [60]. In a brief description, the idea behind the FRBS is to emulate a strategy that penalizes moves (assigning a high value for the tabu tenure) that in the recent past iterations were often present in the tabu list. In opposition, moves that were rare and not recently present in the tabu list received lower values for the tabu tenure. Given the tabu list history, two measures were recorded for each element in the tabu list:

***Frequency* -** its relative frequency, that is, the ratio between the number of times it has entered the tabu list and the actual number of iterations;

***Inactivity*-** the last iteration in which it entered the tabu list.

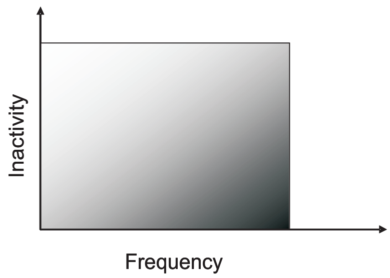

Figure 2: Tabu Tenure depending on Frequency and Inactivity

The idea is to construct a Rule Based System where, for each entering tabu item, giving it corresponding pair (*Frequency,Inactivity*) a rule is fired indicating how long (number of iterations) it will remain in the tabu list, that is, its tabu tenure.
The two concepts (Frequency and Inactivity) were defined as Linguistic Variables [80] with 3 linguist terms each – LOW, MEDIUM and HIGH, and fuzzyfied using a gaussian membership function. Since there are two linguistic variables, with 3 linguistic terms, a total of 9 rules were used to find a value z_i, $i = 1, \ldots, 9$ as shown in Table 2. This value is used to produce an integer value, the tabu tenure, using a 0-order Sugeno system [71]. The details are similar to the implementation in [60].

Rule 1: **IF** *Frequency* is	LOW	**AND** *Inactivity* is HIGH	**THEN** z_1
Rule 2: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is HIGH	**THEN** z_2
Rule 3: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is HIGH	**THEN** z_3
Rule 4: **IF** *Frequency* is	LOW	**AND** *Inactivity* is MEDIUM	**THEN** z_4
Rule 5: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is MEDIUM	**THEN** z_5
Rule 6: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is MEDIUM	**THEN** z_6
Rule 7: **IF** *Frequency* is	LOW	**AND** *Inactivity* is LOW	**THEN** z_7
Rule 8: **IF** *Frequency* is	MEDIUM	**AND** *Inactivity* is LOW	**THEN** z_8
Rule 9: **IF** *Frequency* is	HIGH	**AND** *Inactivity* is LOW	**THEN** z_9

Table 2: Rules in FRBS

To graphically show the behaviour of the Sugeno system, Figure 2 illustrates the pattern of the tabu tenure value according to the values of *Frequency* and *Inactivity.* Darker colours indicate higher values for the tabu tenure. Further details can be found in [60].

3.6. Neighbourhood evaluation and selection

For the current iteration solution, a set of neighbouring points is generated and evaluated. The best point according to some criteria will be chosen as the new starting point for the next iteration. In a single objective problem, its value is often used to rank the solutions. In the multi-objective case, one possibility is to aggregate the several functions into a single objective. In this paper we chose a different approach and a multi-attribute problem was considered. The reason for doing so was to avoid a premature conditioning of the problem, allowing a broader inspection of solutions

in a sort of diversification strategy.
For a given set of points, $T_1, T_2, \ldots, T_r$, their corresponding values of the above mentioned objectives functions f_1, f_2, f_3, f_4 were calculated. To select a point, a multi-attribute decision-making (MADM) problem had to be solved where the points are regarded as alternatives and the value of the objective functions can be viewed as attributes.

3.7. Data Normalisation

Typically, the attributes in multi-attribute problems are normalised since they may be defined according to different measures. To make them comparable, a rescaling of the values is most often performed. One common technique is the division by the maximum, but in some cases it happens that the reference value is a smaller value than the maximum. For instance, when one attribute is price, the user may have a threshold that is lower than the most expensive item. In this case, the utility of an item where price is above the threshold should be set to zero. Our approach was

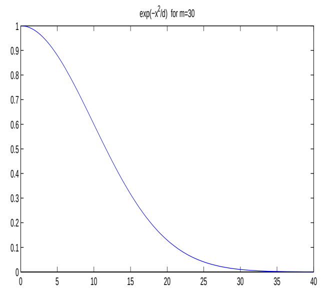

Figure 3: Gaussian mapping function

to map the values on the interval $[0, 1]$, in a sort of fuzzyfication approach. The positive tale of a gaussian function with zero mean and unitary variance was used. This function's smoothness and shape performs a good representation of utility. In this case, the 0 is transformed into the highest possible value of 1. It corresponds to the ideal situation of complete satisfaction of the objective function. For values near the ideal, the penalty is not severe, but there is a deep accentuation of this severity for values higher than an average value. The same expression for the four objectives $f_i(T), i = 1, \ldots, 4$ was used:

$$\widetilde{f}_i(T) = e^{\left(\frac{(f_i(T))^2}{d_i}\right)} \text{ where } d_i = \frac{m_i^2}{\log(10^{-2})}, \tag{9}$$

where the only difference is in the value of the constant m_i for $i = 1, \ldots, 4$. These values were set to

$$m_1 = 0.10M, \tag{10}$$
$$m_2 = 0.15M, \tag{11}$$
$$m_3 = 0.25M, \tag{12}$$
$$m_4 = M, \tag{13}$$

where M is the total number of enrolments for exams. These m_i values are related to a threshold that corresponds to the highest reasonable value above which it is considered an undesirable situation. The value of 10^{-2} in (9) depends on the working precision and corresponds to $\widetilde{f}_i(m_i)$.

Numerical example 1. *Given a set of 100 students and considering $M = 400$, Table 3 depicts an attribute value (f_i), the constant m_i, the value d_i in (9), and the normalised attribute value $\widetilde{f}_i$. Moreover, each row corresponds to different objective function values which can be viewed as attributes.*

Table 3: Normalised values

i	f_i	m_i	d_i	$\widetilde{f}_i$
1	20	40	347.44	0.32
2	61	60	781.73	0.01
3	80	100	2171.47	0.05
4	120	400	34743.56	0.66

This small example helps to understand the effect of constants m_i. In a total of 400 exams enrolments, 40, 60, 100 and 400 represent a threshold for each objective above which the utility is less or equal than 10^{-2}.

4. Compromise Ratio

In each iteration of the TS, the set of V neighbouring points T_i, for $i = 1, \cdots, V$, are evaluated using the four objective functions. After normalising the values, a matrix $\mathbf{X}$ with V alternatives and 4 attributes is obtained.

$$\begin{array}{c} \\ \text{ALTERNATIVES} \left\{ \begin{array}{c} \\ \\ \\ \\ \end{array} \right. \end{array} \begin{array}{c} \mathbf{X} \\ T_1 \\ T_2 \\ \vdots \\ T_V \end{array} \begin{array}{c} \overbrace{\begin{array}{cccc} \widetilde{f}_1 & \widetilde{f}_2 & \widetilde{f}_3 & \widetilde{f}_4 \end{array}}^{\text{ATTRIBUTES}} \\ \left[\begin{array}{cccc} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \cdots & \cdots & \cdots & \cdots \\ x_{V1} & x_{V2} & x_{V3} & x_{V4} \end{array} \right] \end{array}$$

To proceed to the next iteration, it is necessary to choose one of the timetabling solutions from $\{T_1, T_2, \ldots, T_V\}$ which will be set as the next reference solution. If a solution dominates all the others its choice is clear. Domination of a solution T_i with attributes $(x^*_{i1}, x^*_{i2}, x^*_{i3}, x^*_{i4})$ implies that,

$$x^*_{ik} = \max_{j=1,\ldots,V} \{x_{jk}\}, \text{ for } k = 1, 2, 3, 4.$$

Typically, none of these solutions dominate the others. Hence, it is necessary to make a choice based on the 4 criteria already mentioned. Different multi-attribute methods like AHP [68], ELECTRE [67], PROMETHEE [11] can be used to elect a solution. Since this multi-attribute problem must be solved in every iteration, the aforementioned techniques are not suitable. Their inclusion as a subroutine in each iteration of the TS represents a significant computational effort. Other simple methods, such as MaxiMin, MiniMax and Conjuntive&Disjunctive, could be implemented, but they lack some sophistication necessary to produce a good selection strategy, 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 leads to the choice of Compromise Ratio [50]. This method can be viewed as an extension of TOPSIS [45].

The Compromise Ratio is based on the idea that the best alternative should be as close as possible to the ideal solution a^+ (a four-dimensional vector of ones in our case), and as far as possible from the negative-ideal solution a^- (a four-dimensional vector of zeros). If the attributes have different degrees of importance for the decision-maker, a component-wise weighting of matrix $\mathbf{X}_{n\times m} = (x_{ij})$ may be performed,

$$v_{ij} = x_{ij} \times w_{ij}. \tag{14}$$

In general, the weights depend only on the attributes, so $w_{ij} = w_j$. For each point T_i, we needed to compute the distances to the ideal and negative-ideal point, respectively

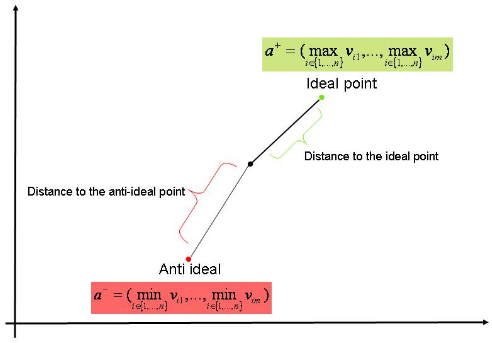

Figure 4: Distance to ideal and negative-ideal point.

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m}(a_j^+ - v_{ij})^p} \quad ,\forall i = 1, ..., n \tag{15}$$

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m}(v_{ij} - a_j^-)^p} \quad ,\forall i = 1, ..., n. \tag{16}$$

Given these distances and in order to rank the alternatives, the following parametrised ratio was used,

$$\xi_p(T_i) = \theta \times \frac{D_{1p}(T^+) - D_p^+(T_i)}{D_{1p}(T^+) - D_{2p}(T^+)} + (1-\theta) \times \frac{D_p^-(T_i) - D_{2p}(T^-)}{D_{1p}(T^-) - D_{2p}(T^-)}, \tag{17}$$

where $\theta \in [0, 1]$, and

$$\begin{cases} D_{1p}(T^+) = \max\limits_{i\in\{1,\dots,n\}} \{D_p^+(T_i)\} \\ D_{2p}(T^+) = \min\limits_{i\in\{1,\dots,n\}} \{D_p^+(T_i)\} \\ D_{1p}(T^-) = \max\limits_{i\in\{1,\dots,n\}} \{D_p^-(T_i)\} \\ D_{2p}(T^-) = \min\limits_{i\in\{1,\dots,n\}} \{D_p^-(T_i)\} \end{cases}.$$

The value of θ reflects the decision-maker's attitude towards the proximity to the ideal and the negative-ideal solution. If θ is close to 1, it means that solutions near the ideal are more valued than solutions away from the negative-ideal. Equal importance to both distances is given by $\theta = 1/2$. Solutions are ranked based on $\xi_p(T_i)$, which measures how alternative T_i complies with the compromise established by θ. Note that if a solution T_i is simultaneously the closest to the ideal solutions and the furthest from the negative-ideal solution, then $\xi_p(T_i) = 1$ attains the maximal value.

4.1. Weighting functions

The weights in (14) are defined by the user, and highly influence the final ranking of the alternatives. Hence, their choice must be careful. An interactive definition of the weights by the user is a drawback since an important feature of timetabling software is automation. In addition, capturing preferences by a simple linear model seems too restrictive. Sometimes, the decision-maker does not react in a linear or independent fashion regarding the attributes. For instance, consider the classical situation of buying a car. The evaluation of alternatives can include price and comfort as main attributes. If the car is expensive, it is also expected to be very comfortable. However, if the car is not expensive, then there is no such expectation. In this situation, the value of one attribute influences the weight given to another attribute. The same happens to the attributes regarding the examination timetabling problem. For instance, if many students have more than one exam on the same day, then it is necessary to reinforce the weight that penalises the occurrence of exams in consecutive periods. In [66], to model similar situations, the use of weighting functions instead of fixed weights was proposed. The gain is that preferences are modelled in a more general way, capturing more complex relations than in a simple linear model. The weights are defined by the following expression,

$$\widetilde{w}_{ij}(x_{ij}) = \frac{g_j(x_{ij})}{\sum_{t=1}^{m} g_t(x_{it})}, \tag{18}$$

where the g_j are weights generating functions.

Mixture operators, in the context of aggregation operators were introduced in [52], and in [66] some interesting applications were presented. In this work, weighting functions are used in the context of Compromise Ratio. By replacing in (14) the weights by weighting functions, a new procedure for ranking the alternatives is proposed which has a more realistic way of modelling preferences.

4.2. Modified Compromise Ratio

If we use weighting functions, then the ideal point a^+ becomes

$$\frac{1}{\sum_{t=1}^{m} g_t(1)} \left(g_1(1), \ldots, g_m(1)\right), \tag{19}$$

and the negative-ideal is the null vector The equations analogue to (15) and (16) are

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(a_j^+ - x_{ij}\widetilde{w}_{ij}(x_{ij})\right)^p} \quad , \forall i = 1, ..., n, \tag{20}$$

$$D_p^-(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(x_{ij}\widetilde{w}_{ij}(x_{ij})\right)^p} \quad , \forall i = 1, ..., n. \tag{21}$$

Now the issue is to choose 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 devised a condition similar to the one presented in [52] and [56].

Theorem 1. *The inequalities*

$$0 \le g_k \le 1, \qquad \forall k \in \{1, \ldots, m\}, \tag{22}$$

$$\frac{\partial g_k}{\partial x_{ik}} \ge 0, \qquad \forall k \in \{1, \ldots, m\}, \tag{23}$$

$$x_{ik}^{p-1} g_k - \frac{\partial g_k}{\partial x_{ik}} \ge 0, \qquad \forall k \in \{1, \ldots, m\}, \tag{24}$$

are sufficient conditions for the monotonicity of ξ_p as defined in (17).

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

$$\xi_p(T_i) = \theta \times \frac{D_{1p}(T^+) - D_p^+(T_i)}{D_{1p}(T^+) - D_{2p}(T^+)} + (1-\theta) \times \frac{D_p^-(T_i) - D_{2p}(T^-)}{D_{1p}(T^-) - D_{2p}(T^-)},$$

then if $\frac{\partial D_p^+}{\partial x_{ik}} \leq 0$, and $\frac{\partial D_p^-}{\partial x_{ik}} \geq 0$, we have $\frac{\partial \xi_p}{\partial x_{ik}} \geq 0$. First it is proved that

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

By definition

$$D_p^+(T_i) = \sqrt[p]{\sum_{j=1}^{m} \left(\frac{g_j(1) - x_{ij} g_j(x_{ij})}{\sum_{t=1}^{m} g_t(x_{it})} \right)^p} \quad , \text{ for } p \geq 1. \tag{25}$$

To simplify (25) consider the expressions:

$$\gamma_q(x_i) = \left(\frac{g_q(1) - x_{iq} g_q(x_{iq})}{\varepsilon(x_i)} \right)^p \text{ for } q = 1, \ldots, m, \tag{26}$$

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

Now

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

and we have

$$\frac{\partial D_p^+}{\partial x_{ik}} = \frac{1}{p} \frac{\frac{\partial \sigma}{\partial x_{ik}}}{(D_p^+(T_i))^{p-1}}. \tag{28}$$

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

$$\frac{\partial \gamma_q}{\partial x_{ik}} = p \left(\frac{g_k(1) - x_{ik} g_k(x_{ik})}{\varepsilon(x_i)} \right)^{p-1} \frac{\left(-x_{ik} \frac{\partial g_k}{\partial x_{ik}} - g_k(x_{ik}) \right) \varepsilon(x_i) - \frac{\partial g_k}{\partial x_{ik}} \left(g_k(1) - x_{ik} g_k(x_{ik}) \right)}{\varepsilon(x_i)^2}. \tag{29}$$

Using the left inequality of hypothesis (22) we may conclude that

$$\varepsilon(x_i) \geq 0. \tag{30}$$

Since $x_{ij} \leq 1$ and using condition (23) we know that

$$g_k(1) - x_{ik} g_k(x_{ik}) \geq 0. \tag{31}$$

From (30) and (31) and taking in consideration that $p \geq 1$, we proved that

$$p \left(\frac{g_k(1) - x_{ik} g_k(x_{iq})}{\varepsilon(x_i)} \right)^{p-1} \geq 0. \tag{32}$$

Since $x_{ij} \geq 0$, and using condition (30), hypotheses (23), and the left side of condition (22) we have

$$\left(-x_{ik} \frac{\partial g_k}{\partial x_{ik}} - g_k(x_{ik}) \right) \varepsilon(x_i) \leq 0. \tag{33}$$

From (31) and (23),

$$-\frac{\partial g_k}{\partial x_{ik}}\left(g_k(1)-x_{ik}g_k(x_{ik})\right)\leq 0. \tag{34}$$

So from (32), (33) and (34) it is proved that

$$\frac{\partial \gamma_q}{\partial x_{ik}}\leq 0. \tag{35}$$

Now, when $q\neq k$ we have

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

and by (32) and (34) we obtain

$$\frac{\partial \gamma_q}{\partial x_{ik}}\leq 0. \tag{36}$$

From (35) and (36)

$$\frac{\partial \sigma}{\partial x_{ik}}\leq 0. \tag{37}$$

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

$$\frac{1}{p(D_p^+(T_i))^{p-1}}\geq 0. \tag{38}$$

Using (28), (37), and (38) we may conclude that

$$\frac{\partial D_p^+}{\partial x_{ik}}\leq 0. \tag{39}$$

Next we prove that

$$\frac{\partial D_p^-}{\partial x_{ik}}\geq 0. \tag{40}$$

First we take in consideration that

$$D_p^-(T_i)=\sqrt[p]{\sum_{j=1}^{m}\left(\frac{x_{ij}g_j(x_{ij})}{\sum_{t=1}^{m}g_t(x_{it})}\right)^p}\quad,\ \text{for } p\geq 1. \tag{41}$$

Again for simplification we introduce the following,

$$\overline{\gamma}_q(x_i)=\left(\frac{x_{iq}g_q(x_{iq})}{\overline{\varepsilon}(x_i)}\right)^p\ \text{ for } q=1,\dots,m \tag{42}$$

where $x_i=(x_{i1},\dots,x_{im})$, $\overline{\varepsilon}(x_i)=\sum_{t=1}^{m}g_t(x_{it})$, and $\overline{\sigma}(x_i)=\sum_{q=1}^{m}\overline{\gamma}_q(x_i)$. Now,

$$\frac{\partial D_p^-}{\partial x_{ik}}=\frac{1}{p}\frac{\frac{\partial\overline{\sigma}}{\partial x_{ik}}}{(D_p^-(T_i))^{p-1}}. \tag{43}$$

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

$$\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} = p\left(\frac{x_{ik} g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i) - \frac{\partial g_k}{\partial x_{ik}}\left(x_{ik} g_k(x_{ik})\right)}{\overline{\varepsilon}(x_i)^2} \tag{44}$$

$$= p\left(\frac{x_{ik} g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} \tag{45}$$

$$-p\left(\frac{x_{ik} g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}\left(x_{ik} g_k(x_{ik})\right)}{\overline{\varepsilon}(x_i)^2}. \tag{46}$$

Using the right inequality of hypothesis (22), and the fact that $x_{ik} \leq 1$ we may conclude that

$$\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} \geq p\left(\frac{x_{ik} g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} \tag{47}$$

$$-p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}\left(x_{ik} g_k(x_{ik})\right)}{\overline{\varepsilon}(x_i)^2}. \tag{48}$$

When $q \neq k$ we obtain

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

Since $g_k(x_{ik}) \leq 1$, and $x_{ik} \leq 1$ we have

$$\frac{\partial \overline{\gamma}_q}{\partial x_{ik}} \geq -p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}\left(x_{ik} g_k(x_{ik})\right)}{\overline{\varepsilon}(x_i)^2}. \tag{49}$$

From (48) and (49) we obtain

$$\frac{\partial \overline{\sigma}}{\partial x_{ik}} \geq p\left(\frac{x_{ik} g_k(x_{ik})}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{\left(x_{ik}\frac{\partial g_k}{\partial x_{ik}} + g_k(x_{ik})\right)\overline{\varepsilon}(x_i)}{\overline{\varepsilon}(x_i)^2} \tag{50}$$

$$-p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}}{\overline{\varepsilon}(x_i)^2} \sum_{s=1}^{m}\left(x_{is} g_s(x_{is})\right). \tag{51}$$

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

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

$$-p\left(\frac{1}{\overline{\varepsilon}(x_i)}\right)^{p-1} \frac{-\frac{\partial g_k}{\partial x_{ik}}}{\overline{\varepsilon}(x_i)^2}\overline{\varepsilon}(x_i), \tag{53}$$

$$\frac{\partial \overline{\sigma}}{\partial x_{ik}} \geq p\frac{1}{\overline{\varepsilon}(x_i)^p} g_k(x_{ik})^{p-1} x_{ik}^p \frac{\partial g_k}{\partial x_{ik}} + p\frac{1}{\overline{\varepsilon}(x_i)^p}\left(x_{ik}^{p-1} g_k(x_{ik})^p - \frac{\partial g_k}{\partial x_{ik}}\right). \tag{54}$$

Given conditions (22), (23), $\overline{\varepsilon}(x_i) \geq 0$, and $p \geq 1$ we obtain

$$p\frac{1}{\overline{\varepsilon}(x_i)^p} g_k(x_{ik})^{p-1} x_{ik}^p \frac{\partial g_k}{\partial x_{ik}} \geq 0. \tag{55}$$

The hypotheses (24), $\overline{\varepsilon}(x_i) \geq 0$, and $p \geq 1$ ensure that

$$p\frac{1}{\overline{\varepsilon}(x_i)^p}\left(x_{ik}^{p-1} g_k(x_{ik})^p - \frac{\partial g_k}{\partial x_{ik}}\right) \geq 0. \tag{56}$$

Finally, from (56) $\frac{\partial D_p^-}{\partial x_{ik}} \geq 0$, which concludes the proof.■

The next issue is the choice of the weighting functions verifying conditions of Theorem 1. Clearly linear functions do not meet the requirements, so other functions are proposed in Theorem 2.

Theorem 2. *The following functions verify the hypotheses (22) to (24) of Theorem 1.*

$$g(x) = (\alpha - \delta)x^p + \delta, \quad (57)$$

$$g(x) = \delta\left(\frac{\alpha}{\delta}\right)^{x^p}, \quad (58)$$

$$g(x) = \log\left(1 + e^{\alpha-\delta} - 1)x^p\right) + \delta. \quad (59)$$

In (57), (58), and (59), the constant α belongs to the unit interval and represents the importance (weight) of the attribute when its satisfaction is maximal. Constant δ represents the importance (weight) of the attribute when its satisfaction is minimal. However, δ must belong to the interval $[lower\_bound, \alpha]$, where *lower_bound* depends on the α value. In the computational experiments, the function (57) and a value of $p = 2$ was used for all attributes. From the conditions presented in Theorem 1 the value of *lower_bound* was set to $\frac{2}{3}\alpha$.

Numerical example 2. *Using the normalisation defined (9), we considered 3 different solutions* T_0, T_1 *and* T_2, $\delta = 0.1$ *and* α_i *for* $i = 1, 2, 3, 4$ *in (57) equal to* $(0.9,\ 0.8\ ,0.7,\ 0.75)$, *respectively. Tables 4 and 5 summarise the results of the weighted Compromise Ratio.*

Table 4: Variation of ponderation by weighting functions for three solutions

$\mathbf{T_0}$	x_{0i}	g_i	$\widetilde{w}_{0i}$	$\widetilde{v}_{0i}$	$\mathbf{T_1}$	x_{1i}	g_i	$\widetilde{w}_{1i}$	$\widetilde{v}_{1i}$	$\mathbf{T_2}$	x_{2i}	g_i	$\widetilde{w}_{2i}$	$\widetilde{v}_{2i}$
3	0.97	0.88	0.32	0.31	10	0.75	0.77	0.30	0.23	10	0.75	0.77	0.32	0.24
20	0.60	0.63	**0.22**	**0.13**	**20**	0.60	0.63	**0.25**	**0.15**	**20**	0.60	0.63	**0.26**	**0.16**
20	0.83	0.67	0.24	0.20	25	0.75	0.64	0.25	0.19	100	0.01	0.50	0.21	0.00
90	0.79	0.61	0.22	0.17	200	0.32	0.49	0.19	0.06	200	0.32	0.49	0.21	0.06

The columns x_{0i}, x_{1i} *and* x_{2i} *in Table 4 represent the normalized values. For each* x_j, *the weighting generation function's values from (57) are given in column* g_i *and the weights as in (18) are represented in column* $\widetilde{w}_{ij}$. *The average values similar to (14) are displayed in columns* $\widetilde{v}_{ji}$. *Analysing the bold values in the second row below the header for the same value of* 20, *we see that the corresponding weight is aggravated in the sequence of the three solutions, influenced by the remaining values. Considering now another set of solutions in Table 5, the performance of the combination of the weighting functions with Compromise Ratio* (ξ_W)*is exemplified in comparison with a classical Compromise Ratio* (ξ_L) *and a linear operator (Lin-Ag) with* α *weights. For the Compromise Ratio, three different values of* θ *(17) are tested.*

Table 5: Comparison of operators

		$\mathbf{T_0} = (3, 30, 40, 77)$	$\mathbf{T_1} = (5, 28, 45, 95)$	$\mathbf{T_2} = (10, 20, 40, 75)$
ξ_W	$\theta = 0.3$	0.71	0.35	0.30
	$\theta = 0.5$	0.51	0.25	0,50
	$\theta = 0.8$	0.22	0.10	0.80
ξ_L	$\theta = 0.3$	0.72	0.00	1,00
	$\theta = 0.5$	0.60	0.00	1.00
	$\theta = 0.8$	0.40	0.00	1,00
Lin-Ag	$\alpha = (0.9, 0.8, 0.7, 0.75)$	0.66	0.62	0.67

The results show that the linear aggregation hardly differentiates the three solutions. It is also clear that the Compromise Ratio with weighting function has a greater ability to capture subtle differences among solutions and is more sensitive towards changes in θ.

5. Computational experience

Table 6: Data set

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

The algorithm proposed in this paper for the multi-objective approach for exams timetabling problem was tested using the Toronto's benchmark data set [24]. The main characteristics of the data instances are displayed in Table 6. The last column D.C.M. represents the density of the conflict matrix. This matrix has 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 non-zero elements represents its density. In general, higher density values increase the difficulty of 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. For some larger instances, we allowed the algorithm to run for 2 hours. The main goal was to test the algorithm for the minimisation of all the objectives and also to understand how the algorithm performs when using different weighting factors - α in (57) (δ was fixed to *lower_bound* which was set to $\frac{2}{3}\alpha$).

Table 7: Computational results - Complete discrimination

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**
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**
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**
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**

The computational results are depicted in Tables 7, 8 and 9. The α-weights for the different objectives are described in the first column of each table as a four-dimensional vector. For instance, in Table 7 for the data yor-f-83, the weights regarding the first, second, third and fourth objective are $(1, 1, 1, 1)$. In each column, the lower value obtained for the tested α-weights is displayed in bold.

Table 8: Computational results - Average discrimination

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(10.01%)	328
(0.2,0.2,0.2,1)	5427	3982	**6021**	**34**
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(8.67%)
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(58.65%)	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(12.21%)
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**	**3271**	1307
(0.2,0.2,1,0.2)	12438	1452	3359(2.69%)	872
(0.2,0.2,0.2,1)	14415	2198	3619	**209**

The results are presented in three tables to distinguish the outcome. In Table 7, a complete discrimination of objectives was obtained in relation to the highest $\alpha-$weight. As expected, high values of α in one objective tend to induce lower values for that objective value at the expense of the remaining objectives. For the data in Table 8, it is still possible to identify similar but less strict behaviour. However, the relative gap $(100\frac{x-\text{best value}}{\text{best value}}\%)$ towards the best known value is still small in most cases. Moreover, considering that this happened just for one objective, it cannot be considered a failure.

Table 9: Computational results - Special cases

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(8.04%)	4262	5896	373
(0.2,1,0.2,0.2)	8074	**4093**	5734	330
(0.2,0.2,1,0.2)	**6753**	4470	6378(13.37%)	339
(0.2,0.2,0.2,1)	8332	4191	**5626**	253(8.58%)
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**(0%)	**5090**	327
(0.2,0.2,1,0.2)	10595	**3929**	**5090**(0%)	327
(0.2,0.2,0.2,1)	11452	4015	5304	**286** (0%)
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**(0%)	3202	4678	1148
(0.2,1,0.2,0.2)	**8422**	3202(0.22%)	4678	1148
(0.2,0.2,1,0.2)	**8422**	3202	4678(5.67%)	1148
(0.2,0.2,0.2,1)	9577	3254	**4427**	**838**

Preliminary results for instances uta-s-92, rye-s-93 and kfu-s-93 with a 1 hour running time limit presented a poor ability to discriminate between objectives. This limit was increased to 2 hours for these three data sets and the quality of the results was improved. A complete discrimination was obtained for uta-s-92 (Table 7) and small gaps for rye-s-93 and kfu-s-93 (Table 8). Table 9 presents the cases where a strict discrimination between objectives was not achieved for two objectives. However, the gaps are still small. In car-s-91, and particularly for sta-f-83, there are many different weights which obtain the same values. One possible explanation is that the regions of attraction of the 4 objectives are very close. This can also be corroborated by the fact that the absolute gaps are also small when the first row is taken as a reference.

It is interesting to note that the performance does not seem to be related to the potential difficulty induced by higher values of D.C.M.. It can also be observed that increasing the running time may be crucial.

6. Conclusions

In this paper, we presented a new approach to solve a multi-objective examination timetabling problem. The difficulty in addressing these problems is well known, and so it is always a challenge to develop more efficient approaches. The use of the Tabu Search is well justified due to the complexity of the problem. In order to increase the automation of the method, a Fuzzy Inference Ruled Based System was developed to manage an appropriate choice of the tabu tenure. In order to evaluate the neighbouring solutions in each iteration, a modified version of the Compromise Ratio multi-attribute method was developed. The replacement of the fixed weights by weighting generating functions was proposed. This change resulted in a more automatic, complex and realistic modelling of preferences by the decision-maker. To guarantee the monotonicity of the aggregation operator, some sufficient conditions were established for the weighting generation function. Moreover, a set of functions were identified which satisfy these conditions. The proposal presented in this paper can be easily adapted to other problems and the theoretical results presented for the modified Compromise Ratio method can be used for any other multi-attribute problem.

References

[1] S. Abdullah, S. Ahmadi, E. K. Burke, and M. Dror. Investigating ahuja–orlinâs large neighbourhood search approach for examination timetabling. *OR Spectrum*, 29(2):351–372, 2007.

[2] S. Abdullah, H. Turabieh, B. McCollum, and P. McMullan. A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics*, 18(1):1–23, 2012.

[3] 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.

[4] 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.

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

[6] H. Asmuni, E. K. Burke, J. Garibaldi, and B. McCollum. Fuzzy multiple heuristic orderings for examination timetabling. In *Practice and Theory of Automated Timetabling V*, pages 334–353. Springer, 2005.

[7] R. M. Awad and J. W. Chinneck. Proctor assignment at carleton university. *Interfaces*, 28(2):58–71, 1998.

[8] M. Banbara, T. Soh, N. Tamura, K. Inoue, and T. Schaub. Answer set programming as a modeling language for course timetabling. *Theory and Practice of Logic Programming*, 13:783–798, 2013.

[9] R. Bellio, L. Di Gaspero, and A. Schaerf. Design and statistical analysis of a hybrid local search algorithm for course timetabling. *Journal of Scheduling*, 15(1):49–61, 2012.

[10] N. Boland, B. D. Hughes, L.T.G. Merlot, and P.J. Stuckey. New integer linear programming approaches for course timetabling. *Computers & Operations Research*, 35(7):2209–2233, 2008.

[11] J. P. Brans and Ph. Vincke. A preference ranking organisation method: (the promethee method for multiple criteria decision-making). *Management Science*, 31(6):647–656, 1985.

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

[13] S. Broder. Final examination scheduling. *Communications of the ACM*, 7(8):494–498, 1964.

[14] 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.

[15] E. Burke, K. Jackson, J.H. Kingston, and R. Weare. Automated university timetabling: The state of the art. *The computer journal*, 40(9):565–571, 1997.

[16] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1):46–53, 2010.

[17] E. K. Burke and J. Newall. Enhancing timetable solutions with local search methods. In *Practice and Theory of Automated Timetabling IV*, pages 195–206. Springer, 2003.

[18] E. K. Burke and J. P. Newall. A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1):63–74, 1999.

[19] E. K. Burke, R. Qu, and J. Li. A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research*, 203(2):484–493, 2009.

[20] Y. Bykov. *Time-Predefined and Trajectory-Based Search: Single and Multiobjective Approaches to Exam Timetabling.* PhD thesis, School of Computer Science & Information Technology - University of Nottingham, 2003.

[21] V. Cacchiani, A. Caprara, R. Roberti, and P. Toth. A new lower bound for curriculum-based course timetabling. *Computers & Operations Research*, 40(10):2466–2477, 2013.

[22] A. Caprara, M. Fischetti, P.L. Guida, M. Monaci, G. Sacco, and P. Toth. Solution of real-world train timetabling problems. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 1057–1066, 2001.

[23] M. Caramia, P. DellÓlmo, and G. Italiano. New algorithms for examination timetabling. *Algorithm Engineering*, pages 230–241, 2001.

[24] M. Carter and G. Laporte. Recent developments in practical examination timetabling. In *Practice and theory of automated timetabling*, pages 1–21, 1996.

[25] M. W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, pages 193–202, 1986.

[26] M. W. Carter, G. Laporte, and J. W. Chinneck. A general examination scheduling system. *Interfaces*, pages 109–120, 1994.

[27] S. Casey and J. Thompson. Grasping the examination scheduling problem. In *Practice and Theory of Automated Timetabling IV*, pages 232–244. Springer, 2003.

[28] A. Colorni, M. Dorigo, and V. Maniezzo. Metaheuristics for high school timetabling. *Computational optimization and applications*, 9(3):275–298, 1998.

[29] P. Corr, B. McCollum, M. McGreevy, and P. McMullan. A new neural network based construction heuristic for the examination timetabling problem. In *Parallel Problem Solving from Nature - PPSN IX*, pages 392–401. Springer, 2006.

[30] D. Costa and A. Hertz. Ants can colour graphs. *Journal of the operational Research Society*, 48(3):295–305, 1997.

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

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

[33] L. Di Gaspero. Recolour, shake and kick: A recipe for the examination timetabling problem. In *Proceedings of the 4th international conference on the practice and theory of automated timetabling*, pages 404–407, 2002.

[34] L. Di Gaspero and A. Schaerf. Tabu search techniques for examination timetabling. *Practice and Theory of Automated Timetabling III*, pages 104–117, 2001.

[35] K. A. Dowsland and J. M. Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438, 2004.

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

[37] K. Easton, G. Nemhauser, and M. Trick. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter Sports scheduling, page 52. Chapman and Hall (CRC), 2004.

[38] W. Erben. A grouping genetic algorithm for graph colouring and exam timetabling. *Practice and Theory of Automated Timetabling III*, pages 132–156, 2001.

[39] G. Filho and L. Lorena. A constructive evolutionary approach to school timetabling. *Applications of Evolutionary Computing*, pages 130–139, 2001.

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

[41] F. Glover and M. Laguna. *Tabu search*, volume 1. Springer, 1998.

[42] C. Gogos, P. Alefragis, and E. Housos. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of operations research*, 194(1):203–221, 2012.

[43] 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.

[44] P. Hansen and N. Mladenović. Variable neighborhood search. *Handbook of metaheuristics*, pages 145–184, 2003.

[45] C. L. Hwang and K. Yoon. *Multiple attribute decision making: methods and applications : a state-of-the-art survey.* Lecture notes in economics and mathematical systems. Springer-Verlag, 1981.

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

[47] S. N . Jat and S. X. Yang. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. *Journal of Scheduling*, 14(6):617–637, 2011.

[48] G. Kendall and N. Hussin. An investigation of a tabu-search-based hyper-heuristic for examination timetabling. *Multidisciplinary Scheduling: Theory and Applications*, pages 309–328, 2005.

[49] D. Landa-Silva and J. H. Obit. Evolutionary non-linear great deluge for university course timetabling. In E Corchado, Wu X, E Oja, A Herrero, and B Baruque, editors, *Hybrid Artificial Intelligence Systems*, volume 5572 of *Lecture Notes in Artificial Intelligence*, pages 269–276. 4th International Workshop on Hybrid Artificial Intelligence Systems, 2009.

[50] D. F. Li. Compromise ratio method for fuzzy multi-attribute group decision making. *Applied Soft Computing*, 7(3):807–817, 2007.

[51] Z. P. Lu, J. K. Hao, and F. Glover. Neighborhood analysis: a case study on curriculum-based course timetabling. *Journal of Heuristics*, 17(2):97–118, 2011.

[52] R. A. Marques Pereira and G. Pasi. On non-monotonic aggregation: mixture operators. In *Proceedings of the 2nd International Conference on Soft and Intelligent Computing*, pages 513–517, 1999.

[53] B. McCollum, P. McMulla, A. J. Parkes, E. K. Burke, and S. Abdullah. An extended great deluge approach to the examination timetabling problem. In *Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2009)*, pages 424–434, 2009.

[54] B. McCollum, P. McMulla, A. J. Parkes, E. K. Burke, and R. Qu. A new model for automated examination timetabling. *Annals of operations research*, 194(1):291–315, 2012.

[55] L. Merlot, N. Boland, B. Hughes, and P. Stuckey. A hybrid algorithm for the examination timetabling problem. *Practice and theory of automated timetabling IV*, pages 207–231, 2003.

[56] R. Mesiar, J. Spirková, and L. Vavríková. Weighted aggregation operators based on minimization. *Information Sciences*, 178(4):1133–1140, 2008.

[57] S. A. MirHassani and F. Habibi. Solution approaches to the course timetabling problem. *Artificial Intelligence Review*, 39(2):133–149, 2013.

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

[59] C.A. Morgenstern. *Algorithms for general graph coloring.* PhD thesis, University of New Mexico, 1990.

[60] T. C. Pais and P. Amaral. Managing the tabu list length using a fuzzy inference system: an application to examination timetabling. *Annals of Operations Research*, pages 1–23, 2012.

[61] L. F. Paquete and C. M. Fonseca. A study of examination timetabling with multiobjective evolutionary algorithms. In *4th metaheuristics international conference (MIC 2001)*, pages 149–154, 2001.

[62] S. Petrovic and Y. Bykov. A multiobjective optimisation technique for exam timetabling based on trajectories. *Practice and Theory of Automated Timetabling IV*, pages 181–194, 2003.

[63] P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks. Stochastic optimisation timetabling tool for university course scheduling. *International Journal of Production Economics*, 112(2):903–918, 2008.

[64] X. Qi, J. Yang, G. Y. Qi, J. Yang, and G. Yu. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter Scheduling problems in the airline industry, page 50. Chapman and Hall, 2004.

[65] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1):55–89, 2009.

[66] R.A. Ribeiro and R.A. Marques Pereira. Generalized mixture operators using weighting functions: A comparative study with wa and owa. *European Journal of Operational Research*, 145(2):329–342, 2003.

[67] B. Roy. Classement et choix en présence de points de vue multiples. *Recherche opérationnelle*, 2(6):57–75, 1968.

[68] T.L. Saaty. *Analytic hierarchy process*. Wiley Online Library, 1980.

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

[70] 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.

[71] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy sets and systems*, 28(1):15–33, 1988.

[72] I. X. Tassopoulos and G. N. Beligiannis. A hybrid particle swarm optimization based algorithm for high school timetabling-problems. *Applied Soft Computing Journal*, 12(11):3472–3489, 2012.

[73] 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.

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

[75] M. Trick. A schedule-then-break approach to sports timetabling. *Practice and Theory of Automated Timetabling III*, pages 242–253, 2001.

[76] H. Turabieh, S. Abdullah, and B. McCollum. Electromagnetism-like mechanism with force decay rate great deluge for the course timetabling problem. In P Wen, Y Li, L Polkowski, Y Yao, S Tsumoto, and G Wang, editors, *Rough Sets and Knowledge Technology*, volume 5589 of *Lecture Notes in Artificial Intelligence*, pages 497–504. 4th International Conference on Rough Sets and Knowledge Technology, 2009.

[77] G. White and B. Xie. Examination timetables and tabu search with longer-term memory. *Practice and Theory of Automated Timetabling III*, pages 85–103, 2001.

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

[79] Y. Yang and S. Petrovic. A novel similarity measure for heuristic selection in examination timetabling. *Practice and Theory of Automated Timetabling V*, pages 247–269, 2005.

[80] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, part i. *Information Sciences*, 8(3):199–249, 2005.