A Novel Fuzzy Approach to Evaluate the Quality of Examination Timetabling

Hishammuddin Asmuni[1], Edmund K. Burke[1], Jonathan M. Garibaldi[1], and Barry McCollum[2]

[1] School of Computer Science and Information Technology,
University of Nottingham, Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB, UK
{hba,ekb,jmg}@cs.nott.ac.uk

[2] School of Computer Science,
Queen's University Belfast,
Belfast BT7 1NN, UK
b.mccollum@qub.ac.uk

Abstract. In this paper we introduce a new fuzzy evaluation function for examination timetabling. We describe how we employed fuzzy reasoning to evaluate the quality of a constructed timetable by considering two criteria, the average penalty per student and the highest penalty imposed on any of the students. A fuzzy system was created based on a series of easy to understand rules to combine the two criteria. A significant problem encountered was how to determine the lower and upper bounds of the decision criteria for any given problem instance, in order to allow the fuzzy system to be fixed and, hence, applicable to new problems without alteration. In this work, two different methods for determining boundary settings are proposed. Experimental results are presented and the implications analysed. These results demonstrate that fuzzy reasoning can be successfully applied to evaluate the quality of timetable solutions in which multiple decision criteria are involved.

Keywords- Examination timetabling, fuzzy reasoning, multi-criteria decision making

1 Introduction

Timetabling refers to the process of allocating limited resources to a number of events subject to many constraints. Constraints are divided into two types: hard and soft. Hard constraints cannot be violated in any circumstances. Any timetable solution that satisfies all the hard constraints specified is considered as a feasible solution, provided that all the events are assigned to a time slot. Soft constraints are highly desirable to satisfy, but it is acceptable to breach these types of constraint. However, it is very important to minimise the violation of the soft constraints, because, in many cases, the quality of the constructed timetable is evaluated by measuring the fulfillment of these constraints. In practice, the variety of constraints which are imposed by academic institutions are very different [6]. Such variations make the timetabling problem more challenging. Algorithms or approaches that have been successfully applied to one problem, may not perform well when applied to different timetabling instances.

Researchers have employed many different approaches over the years in an attempt to generate 'optimal' timetabling solutions subject to a list of constraints. Approaches such as Evolutionary Algorithms [7, 10, 17, 25], Tabu Search [8, 18, 20, 26], Simulated Annealing [24], Constraint Programming [1, 4, 19], Case Based Reasoning [11, 27] and Fuzzy Methodologies [3, 22, 27] have been successfully applied to timetabling problems.

In 1996, Carter *et al.* [14] introduced a set of examination timetabling benchmark data. This benchmark data set consists of 13 problem instances. Originally these data came from real university examination timetabling problems. Therefore, it was expected that these data sets varied considerably in terms of resources given/availability, constraints specified and how the quality of the constructed timetable were evaluated. For the sake of generality, these data sets were then simplified such that only the following constraints were considered:

Hard constraint The constructed timetable must be conflict free. The requirement is to avoid any student being scheduled for two different exams at the same time.

Soft constraint The solution should attempt to minimise the number of exams assigned in adjacent time slots in such a way as to reduce the number of students sitting exams in close proximity.

In the context of these benchmark data sets, several different objective functions have been introduced in order to measure the quality of the timetable solution. In addition to the commonly used objective function that evaluates only the proximity cost (see next section for details), other objective functions have been derived based on the satisfaction of other soft constraints, such as minimising consecutive exams in one day or overnight, assigning large exams to early time slot, and others. This is discussed in more detail in the following section.

Previous studies such as [3] and [22], demonstrated that fuzzy reasoning is a promising technique that can be used both for modeling timetabling problems and for constructing solutions. These studies indicated that the utilisation of fuzzy methodologies in university timetabling is an encouraging research topic. In this paper, we introduce a new evaluation function that is based on fuzzy methodologies. The research presented in this paper will focus on evaluating the constructed timetable solutions by considering two decision criteria. Although the constructed timetable solutions were developed based on specific objectives specified earlier, the method is general in the sense that a user could, in principle, define additional criteria he or she wished to be taken into account in evaluating any constructed timetables. This paper is motivated by the fact that in practice the quality of the timetable solution is usually assessed by the timetabling officer considering several criteria/objectives.

In the next section, we present a brief description of existing evaluation methods, their drawbacks, and a detailed explanation of the proposed novel approach. Section 3 presents descriptions of the experiments carried out and the results obtained, followed by discussions in Section 4. Finally, some concluding comments and future research directions are given in Section 5.

2 Assessing Timetable Quality

2.1 Existing Evaluation Function

This section presents several evaluation functions that have been developed for Carter *et al.*'s benchmark data sets. The proximity cost function was the first evaluation function used to measure the quality of timetables [14]. It is motivated by the goal of spreading out each student's examination schedule. In the implementation of the proximity cost, it is assumed that the timetable solution satisfies the defined hard constraint i.e. that no student can attend more than one exam at the same time. In addition, the solution must be developed in such a way that it will promote the spreading out of each student's exams so that students have as much time as possible between exams. If two exams scheduled for a particular student are t time slots apart, a penalty weight is set to $w_t = 2^{5-t}$ where $t \in \{1, 2, 3, 4, 5\}$ (as implemented in [14] and widely adopted by most subsequent research in this area). The weight is multiplied by the number of students that sit both the scheduled exams. The average penalty per student is calculated by dividing the total penalty by the total number of students. The maximum number of time slots for each data set are predefined and fixed, but no limitation in terms of capacity per time slot is set. Consecutive exams either in the same day or overnight are treated the same, and there is no consideration of weekends or other actual gaps between logically consecutive days. Hence, the following formulation is used to measure this proximity cost (adapted from Burke *et al.* [5]):

$$\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} w_{|p_j - p_i|}}{S},$$

where N is the number of exams, s_{ij} is the number of students enrolled in both exam i and j, p_i is the time slot where exam i is scheduled, and S is the total number of students; subject to $1 \leq |p_j - p_i| \leq 5$.

Burke *et al.* [10] devised a new evaluation function in which the goal is to minimise the number of students who have to sit two exams in the same day. Besides the need to construct a conflict free timetable, it also required to schedule the exams within the maximum number of time slots given. There are three time slots per weekday and one morning slot on Saturday. A maximum capacity per time slot is also specified. Burke and Newall [9] extended the previous evaluation function by defining different weights for two consecutive exams in the same day and two exams in overnight consecutive time slots.

More recently, Petrovic *et al.* [22] employed fuzzy methodologies to measure the satisfaction of various soft constraints. The authors described how they modeled two soft constraints, namely *constraint on large exam* and *constraint on proximity of exams*, in the form of fuzzy linguistic terms and defined the related rule set. They used these two criteria to evaluate the timetable quality.

2.2 Disadvantages/Drawbacks of Current Evaluation Functions

As can be seen, the final value of the proximity cost penalty function is a measure only of the average penalty per student. Although this penalty function has been widely used by many researchers in the context of the benchmark data set, in practice, considering only the average penalty per student is not sufficient to evaluate the quality of the constructed timetable. The final value does not, for example, represent the relative fairness of spreading out each student's schedule. For example, when examining the resultant timetable, it may be the case that a few students have an examination timetable in which many of their exams are scheduled in adjacent time slots. These students will not be happy with their timetable as they will not have enough time to do their preparation. On the other hand, the remaining students enjoy a 'good' examination timetable.

EXAMPLE : Consider two cases. *Case 1*: there are 100 students with each student given 1 penalty cost; *Case 2*: there are 100 students, but now 10 students are given 10 penalty cost respectively; the rest zero. In both cases the average penalty per student is equal to 1, but obviously the solution in Case 2 is 'worse' than the solution in Case 1.

One of the authors (McCollum), with extensive experience of real-world timetabling, having spend 12 years as a timetabliong officer and with continuing links with the timetabling industry, has expressed (via private communication) that 'proximity cost' is not the only factor considered by timetabling officers when evaluating the quality of a timetable. Usually, a timetable evaluation is based on several factors and some of the factors are subjective and/or based on ambiguous information. Furthermore, to the best of our knowledge, all the evaluation functions mentioned in Section 2.1 are integrated into the timetabling construction process. These objective functions are used to measure the satisfaction of specific soft constraints. This means that, the constructed timetable is optimised for certain soft constraints. In practice, the user may consider other criteria in evaluating the constructed timetable.

One way to handle multiple criteria decision making is by using simple linear combination. This works by multiplying the value of each criterion by a constant weighting factor and summing to form an overall result. Each weight represents the relative important of each criterion compared to the other criteria. In reality, there is no simple way to determine the precise values for these weights, especially weights that can be used across several problem instances with different complexity. Fuzzy systems are a generalisation of a linear system, in that they can implement both linear and non-linear combinations. The nature of fuzzy systems that allows the use of linguistic terms to express the systems' behaviours provides a transparent representation of the nonlinear system under

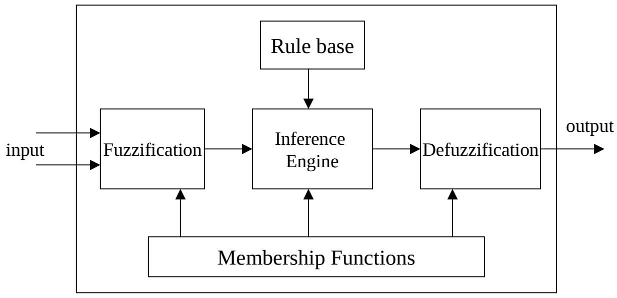


Fig. 1. Components of a fuzzy system

consideration. Fuzzy systems apply '*if-then*' rules and logical operators to map the relationships between input and output variables in the system. Fuzzy rules may be elicited from 'experts', which for the problem under consideration refers to timetabling officers or timetabling consultants. As mentioned earlier, we have access to such experts who could provide us with enough knowledge to develop a fuzzy system.

Therefore, in this paper a new evaluation function utilising fuzzy methodologies is introduced. Basically, the idea is to develop an independent evaluation function that can be used to measure the quality of any constructed examination timetable. The timetable can be generated using any approach with specific objectives to achieve. Subsequently, the timetable solution with the problem description and the list of factors that need to be evaluated are submitted to the evaluation function.

2.3 Overview of Fuzzy Systems

This subsection is largely reproduced from our paper [3] for the purpose of completeness. In many decision making environments, it is often the case that several factors are simultaneously taken into account. Often, it is not known which factor(s) need to be emphasised more in order to generate a better decision. Somehow a trade off between the various (potentially conflicting) factors must be made. The general framework of fuzzy reasoning facilitates the handling of such uncertainty.

Fuzzy systems are used for representing and employing knowledge that is imprecise, uncertain, or unreliable. Figure 1 shows the 5 interconnected components of a fuzzy system. The fuzzification component computes the membership grade for each crisp input variables based on the membership functions defined. The inference engine then conducts the fuzzy reasoning process by applying the appropriate fuzzy operators in order to obtain the fuzzy set to be accumulated in the output variable. The defuzzifier transforms the output fuzzy set to crisp output by applying specific defuzzification method.

More formally, a fuzzy set A of a universe of discourse X (the range over which the variable spans) is characterised by a *membership function* $\mu_A : X \rightarrow [0, 1]$ which associates with each element x of X a number $\mu_A(x)$ in the interval $[0, 1]$, with $\mu_A(x)$ representing the *grade of membership* of x in A [28]. The precise meaning of the membership grade is not rigidly defined, but is supposed to capture the 'compatibility' of an element to the notion of the set. Rules which connect input variables to output variables in 'IF ... THEN ...' form are used to describe the desired system response in terms of *linguistic* variables (words) rather than mathematical formulae. The 'IF' part of the rule is referred to as the 'antecedent', the 'THEN' part is referred to as the 'consequent'. The number of rules depends on the number of inputs and outputs, and the desired behaviour of the system. Once the rules have been established, such a system can be viewed as a non-linear mapping from inputs to outputs.

There are many alternative ways in which this general fuzzy methodology can be implemented in any given problem. In our implementation, the standard Mamdani style fuzzy inference was used with standard Zadeh (min-max) operators. In Mamdani inference [21], rules are of the following form:

$$R_i : \text{if } (x_1 \text{ is } A_{i1}) \text{ and ... and } (x_r \text{ is } A_{ir}) \text{ then } (y\ is\ C_i) \text{ for } i = 1, 2, ..., L$$

where L is the number of rules, x_j $(j = 1, 2, 3, ..., r)$ are input variables, y is output variable, and A_{ij} and C_i are fuzzy sets that are characterised by membership functions $A_{ij}(x_j)$ and $C_i(y)$, respectively. The final output of a Mamdani system is one or more arbitrarily complex fuzzy sets which (usually) need to be defuzzified. It is not appropriate to present a full description of the functioning of fuzzy systems here; the interested reader is referred to Cox [16] for a simple treatment or Zimmerman [29] for a more complete treatment.

2.4 The Proposed Fuzzy Evaluation Function

As an initial investigation, this proposed approach was implemented on solutions which were generated based on the proximity cost requirements (*average penalty*), with one additional factor/objective. Beside the average penalty per student, the highest penalty that occurred amongst the students (*highest penalty*) was also taken into account. However, the latter factor was only evaluated after the timetable was constructed. That is to say, there was no attempt to include this factor in the process of constructing the timetable.

A fuzzy system with these two input variables (*average penalty* and *highest penalty*) and one output variable (*quality*) was constructed. Each of the input variables were associated with three linguistic terms; fuzzy sets corresponding to a meaning of *low*, *medium* and *high*. In addition to these three linguistic terms, the output variable (*quality*) has two extra terms that correspond to meanings of *very low* and *very high*. These terms were selected as they were deemed the simplest possible to adequately represent the problem. Gaussian functions of the form $e^{-(x-c)^2/\sigma^2}$, where c and σ are constants, are used to define the fuzzy set for each linguistic term. This is on the basis that they are the simplest and

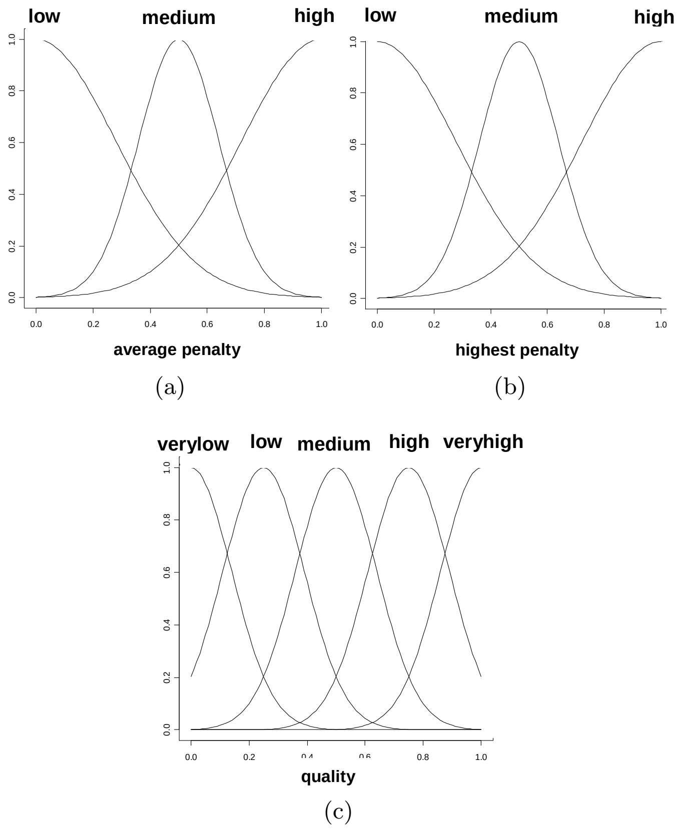


Fig. 2. Membership functions for input and output variables

most common choice, given that smooth, continuously varying functions were desired. The membership functions defined for the two inputs, *average penalty* and *highest penalty*, and the output *quality* are depicted in Figure 2 (a) – (c), respectively.

In the case of such a system having two inputs with three linguistic terms there are nine possible fuzzy rules that can be defined in which each input variable has one linguistic term. As we already know, from the definition of proximity cost, the objective is to minimise the penalty cost, meaning that, the lower the penalty cost, the better the timetable quality. Also, based on everyday experience, we would prefer the highest penalty for any one student to be as low as possible, as this will create more fair timetable for all students. Based upon this knowledge we defined a fuzzy rule set consisting of all 9 possible combinations. Each rule set connects the input variables to a single output variable, *quality*. The fuzzy rule set is presented in Figure 3. As stated above, standard Mamdani style fuzzy inference was used to obtain the fuzzy output for a given set of

Rule 1: IF (*average penalty* is *low*) AND (*highest penalty* is *low*)
THEN (*quality* is *very high*)
Rule 2: IF (*average penalty* is *low*) AND (*highest penalty* is *medium*)
THEN (*quality* is *high*)
Rule 3: IF (*average penalty* is *low*) AND (*highest penalty* is *high*)
THEN (*quality* is *medium*)
Rule 4: IF (*average penalty* is *medium*) AND (*highest penalty* is *low*)
THEN (*quality* is *high*)
Rule 5: IF (*average penalty* is *medium*) AND (*highest penalty* is *medium*)
THEN (*quality* is *medium*)
Rule 6: IF (*average penalty* is *medium*) AND (*highest penalty* is *high*)
THEN (*quality* is *low*)
Rule 7: IF (*average penalty* is *high*) AND (*highest penalty* is *low*)
THEN (*quality* is *medium*)
Rule 8: IF (*average penalty* is *high*) AND (*highest penalty* is *medium*)
THEN (*quality* is *low*)
Rule 9: IF (*average penalty* is *high*) AND (*highest penalty* is *high*)
THEN (*quality* is *very low*)

Fig. 3. Fuzzy rules for *Fuzzy Evaluation System*

inputs. The most common form of defuzzification, 'centre of gravity defuzzification', was then used to obtain a single crisp (real) value for the output variable. This process is based upon the notion of finding the centroid of a planar figure, as given by:

$$\sum_i \frac{\mu(x_i) \cdot x_i}{\mu(x_i)}$$

This single crisp output was then taken as the *quality* of the timetable.

2.5 Input Normalisation

With this proposed fuzzy evaluation function, we carried out experiments to determine whether the fuzzy evaluation system was able to distinguish a range of timetable solutions based on the average penalty per student and the highest penalty imposed on any of the students. All the constructed timetables for the given problem instance were evaluated using the same fuzzy system, and their quality determined based on the output of the fuzzy system. The constructed timetable with the biggest output value was selected to be the 'best' timetable.

Based on our previous experience [2, 3], the average penalty values for different data sets result in widely different scales due to the different complexity of the problem instances. For example, in the STA-F-83 data set (from Carter *et al.*— see below for full details of the data sets used) an average penalty of 160.42 was obtained, whereas for UTA-S-92, the average penalty was 3.57.

As can be seen in Figure 2(a) and Figure 2(b), the input variables have their universe of discourse defined between 0.0 and 1.0. Therefore, in order to use this fuzzy model, both of the original input variables must be normalized within the range $[0.0, 1.0]$. The transformation used is as follows:

$$v' = \frac{(v - lowerBound)}{(upperBound - lowerBound)}$$

where v is the actual value in the initial range $[lowerBound, upperBound]$. In effect, the range $[lowerBound, upperBound]$ represents the actual lower and upper boundaries for the fuzzy linguistic terms.

By applying the normalisation technique, the same fuzzy model can be used for any problem instance, either for the benchmark data sets as used here, or for a new real-world problem. This would provide flexibility when problems of various complexity are presented to the fuzzy system. In such a scheme, the membership functions do not need to be changed from their initial shapes and positions. In addition, rather than recalculate the parameters for each input variable's membership functions, it is much easier to transform the crisp input values into normalized values in the range of $[0.0, 1.0]$. The problem thus becomes one of finding suitable lower and upper bounds for each problem instance.

3 Experiments on Benchmark Problems

3.1 Experiments Setup

In order to test the fuzzy evaluation system, the Carter *et al.*'s [14] benchmark data sets were used. The 12 instances in these benchmark data sets, with different characteristics and various level of complexity, are shown in Table 1.

Table 1. Examination timetabling problem characteristics

Data Set	Number of slots(T)	Number of exams (N)	Number of students (S)
CAR-F-92	32	543	18419
CAR-S-91	35	682	16925
EAR-F-83	24	190	1125
HEC-S-92	18	81	2823
KFU-S-93	20	461	5349
LSE-F-91	18	381	2726
RYE-F-92	23	486	11483
STA-F-83	13	139	611
TRE-S-92	23	261	4360
UTA-S-92	35	622	21266
UTE-S-92	10	184	2750
YOR-F-83	21	181	941

For each instance of the 12 data sets, 40 timetable solutions were constructed using a simple sequential constructive algorithm with backtracking, as previously implemented in [3]. We used 8 different heuristics to construct the timetable solutions, for each of which the algorithm was run 5 times to obtain a range of solutions. However, due to the nature of the heuristics used, in some case, a few of the constructed timetable solutions have the same proximity cost value. Therefore, for the purpose of standardization, 35 different timetable solutions were selected out of the 40 constructed timetable solutions, by firstly removing any repeated solution instances and then just removing at random any excess. The idea is to obtain a set of timetable solutions with variations of timetable solution quality, in which none of the solutions have the same quality in terms of proximity cost (i.e average penalty per student). The timetable solutions were constructed by implementing the following heuristics:

- Three different single heuristic orderings:
  - Least Saturation Degree First (SD),
  - Largest Degree First (LD), and
  - Largest Enrollment First (LE),
- Three different fuzzy multiple heuristic orderings:
  - a *Fixed Fuzzy LD+LE Model*,
  - a *Tuned Fuzzy LD+LE Model*, and
  - a *Tuned Fuzzy SD+LE Model* (see [3] for details of these), and
- random ordering, and
- deliberately 'poor' ordering (see below).

A specific 'poor' heuristic was utilised in an attempt to purposely construct bad solutions. The idea was to attempt to determine the upper bound of solution quality (in effect, though not formally, the 'worst' timetable for the given problem instance). Basically the method was to deliberately assign student exams in adjacent time slots. In order to construct bad solutions, the LD was initially employed to order the exams. Next, the exams were sequentially selected from this ordered exams list, and assigned to the time slot that caused the highest proximity cost; this process continued until all the exams were scheduled.

The 35 timetable solutions were analysed in order to determine the minimum and the maximum values for both the input variables, *average penalty* and *highest penalty*. These values were then used for the normalisation process (see Section 2.5). However, because the 12 data sets have various complexity (see Table 1), the determination of the initial range for each data set is not a straight-forward process. Thus, two alternative boundary settings were implemented in order to identify the appropriate set of *lowerBound* and *upperBound* for each data set.

The first boundary setting used $lowerBound = 0.0$ and the $upperBound = maxValue$, where $maxValue$ is the largest value obtained from the set of 35 solutions. However, from the literature, the lowest value yet obtained for the *STA-F-83* data set is around 130 [15]. Thus, it did not seem sensible to use zero as the lower bound in this case. In order to attempt to address this, we investigated the use of a non-zero lower bound. Of course, a formal method for determining the lower bound for any given timetabling instance is not currently known. Hence, the second boundary setting used *lowerBound* = *minValue* and *upperBound* = *maxValue*, where *minValue* is the smallest value obtained from the set of 35 constructed solutions for the respective data set.

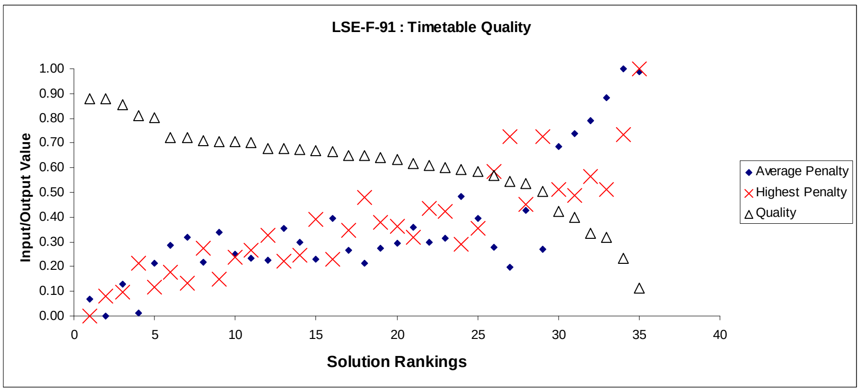
In this implementation, both input variables, *average penalty* and *highest penalty*, were independently normalised based on their respective *minValue* and *maxValue*. The fuzzy evaluation system described earlier (see Section 2.4) was then employed to evaluate the timetable solutions. The same processes were applied to all of the data sets listed in Table 1. The fuzzy evaluation system was implemented using the 'R' language (*The R Foundation for Statistical Computing Version 2.2.0*) [23].

3.2 Experimental Results

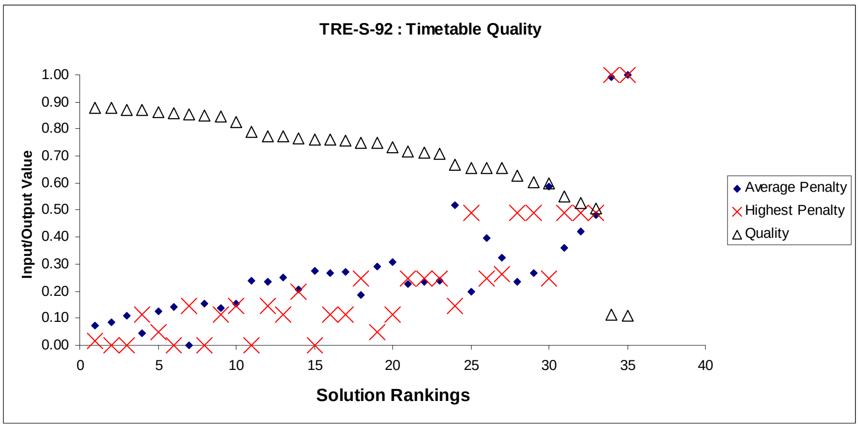
In this section the experiment results are presented. Table 2 shows the minimum and maximum values obtained for both criteria. Figures 4(a) and 4(b) show the evaluation results obtained by the fuzzy evaluation system for the *LSE-F-91* and *TRE-S-92* data sets. These two data sets are shown as representative examples chosen at random. Both graphs show the results obtained when the boundary setting $[minValue, maxValue]$ was implemented. In the graph, the x-axis (Solution Rankings) represents the ranking of the timetable solution quality evaluated by using the fuzzy evaluation function; in the order of the best solution to the worst solution. The y-axis represents the normalised input values (*average penalty* and *highest penalty*) and the output values (*quality*) obtained for the particular timetable solution. These two graphs show that the fuzzy evaluation function has performed as desired, in that the overall (fuzzy) quality of the solutions varies from close to zero to close to one.

Table 2. Minimum and maximum values for *Average Penalty* and *Highest Penalty* obtained from the 35 timetable solutions for each data set

Data Set	*Average Penalty* Minimum Value	*Average Penalty* Maximum Value	*Highest Penalty* Minimum Value	*Highest Penalty* Maximum Value
CAR-F-92	4.54	11.42	65.0	132.0
CAR-S-91	5.29	13.33	68.0	164.0
EAR-F-83	37.02	71.28	105.0	198.0
HEC-S-92	11.78	31.88	75.0	136.0
KFU-S-93	15.81	43.40	98.0	191.0
LSE-F-91	12.09	32.38	78.0	191.0
RYE-F-92	10.38	36.71	87.0	191.0
STA-F-83	160.75	194.53	227.0	284.0
TRE-S-92	8.67	17.25	68.0	129.0
UTA-S-92	3.57	8.79	63.0	129.0
UTE-S-92	28.07	56.34	83.0	129.0
YOR-F-83	39.80	64.48	228.0	331.0



(a)



(b)

Fig. 4. Indicative illustrations of the range of normalized inputs and associated output obtained for the *LSE-F-91* and *TRE-S-92* data sets

Tables 3 and 4 show a comparison of the results obtained using the two alternative forms of the normalisation process. The *Solution Number* is used to identify a particular solution within the 35 timetable solutions used in the experiments for each data set. In both tables, the fifth and sixth columns (labeled as 'Range $[minValue, maxValue]$' indicates the fuzzy evaluation value and the rank of the solution relative to the other solutions, when the boundary range $[minValue, maxValue]$ was used. The last two columns in the tables show the evaluation values and solution ranking obtained when the boundary range $[0, maxValue]$ was used. Only the first 10 'best' timetable solutions for each of the data sets are presented, based on the ranking produced when the boundary range $[minValue, maxValue]$ was used.

Table 3. A comparison of the results obtained using the two alternative forms of the normalisation process for six of the data sets

Data Set	Timetable Criteria			Range[minValue, maxValue]		Range[0, maxValue]	
	Solution Number	Average Penalty	Highest Penalty	Evaluation Value	Solution Ranking	Evaluation Value	Solution Ranking
CAR-F-92	19	4.544	65	0.888503	1	0.534427	1
	17	4.624	71	0.876804	2	0.517946	2
	18	4.639	71	0.876791	3	0.517485	3
	16	4.643	71	0.876788	4	0.517366	4
	7	5.148	68	0.876583	5	0.510084	5
	10	5.192	69	0.873279	6	0.506692	6
	13	5.508	68	0.858276	7	0.500729	7
	12	5.532	68	0.856617	8	0.500120	8
	11	5.595	68	0.851966	9	0.498538	9
	2	5.609	68	0.850863	10	0.498184	10
CAR-S-91	17	5.292	68	0.888524	1	0.557585	1
	13*	5.573	75	0.880205	2	0.537593	3
	11*	5.911	68	0.879621	3	0.542750	2
	15	5.654	75	0.879244	4	0.535472	4
	14	5.842	75	0.875877	5	0.530812	5
	6*	6.079	76	0.868161	6	0.523516	8
	2*	6.393	71	0.860211	7	0.526116	6
	21*	6.509	71	0.853145	8	0.523572	7
	12	5.688	83	0.850233	9	0.520297	9
	16	5.690	83	0.850227	10	0.520255	10
EAR-F-83	21	37.018	116	0.868135	1	0.467867	1
	4*	41.860	118	0.834883	2	0.444700	3
	5*	43.637	105	0.827016	3	0.454672	2
	18	44.147	118	0.798099	4	0.432416	4
	1	41.324	131	0.748303	5	0.415267	5
	3*	43.628	129	0.733864	6	0.411292	7
	20*	44.968	127	0.718542	7	0.411481	6
	12	49.662	114	0.710776	8	0.392966	8
	2*	41.178	144	0.699109	9	0.370814	11
	16*	44.980	135	0.674252	10	0.385906	9
HEC-S-92	21	11.785	83	0.863057	1	0.506506	1
	14	14.774	75	0.854699	2	0.495547	2
	13	13.236	84	0.853706	3	0.489407	3
	7*	14.162	83	0.847966	4	0.482514	5
	16*	14.635	83	0.838633	5	0.477754	7
	15*	14.217	85	0.832653	6	0.476641	8
	1*	15.594	78	0.828916	7	0.481021	6
	6*	15.911	75	0.817611	8	0.485117	4
	27	15.763	84	0.801080	9	0.463727	9
	8*	14.124	94	0.727535	10	0.446459	11
KFU-S-93	17	15.813	98	0.888529	1	0.541211	1
	15	16.904	101	0.884358	2	0.526210	2
	14	17.336	100	0.883340	3	0.524294	3
	16	17.920	104	0.876034	4	0.513226	4
	3*	20.022	102	0.852341	5	0.501383	11
	9*	16.463	113	0.847871	6	0.509402	5
	7*	16.471	113	0.847868	7	0.509339	6
	6*	16.500	113	0.847858	8	0.509119	7
	8*	16.500	113	0.847858	9	0.509119	8
	10*	16.500	113	0.847858	10	0.509119	9
LSE-F-91	11*	13.458	78	0.881499	1	0.552817	2
	13*	12.094	87	0.879126	2	0.555747	1
	6*	14.720	89	0.855424	3	0.523229	4
	12*	12.349	102	0.812127	4	0.527563	3
	10*	16.408	91	0.804048	5	0.504874	5
	32*	17.942	98	0.722929	6	0.480142	7
	5*	18.564	93	0.720053	7	0.481747	6
	9*	16.486	109	0.707889	8	0.476028	9
	16*	18.979	95	0.707212	9	0.474395	11
	7*	17.174	105	0.704871	10	0.476479	8

Table 4. A comparison of the results obtained using the two alternative forms of the normalisation process for the remaining six data sets

Data Set	Timetable Criteria Solution Number	Average Penalty	Highest Penalty	Range[minValue, maxValue] Evaluation Value	Solution Ranking	Range[0, maxValue] Evaluation Value	Solution Ranking
RYE-F-92	21	10.384	87	0.888528	1	0.610225	1
	8	12.180	97	0.871582	2	0.558378	2
	10	12.337	97	0.870489	3	0.556102	3
	20	12.264	98	0.868672	4	0.555205	4
	6	12.976	97	0.864830	5	0.547756	5
	9	12.417	102	0.854386	6	0.545595	6
	7	12.094	105	0.839576	7	0.544225	7
	3*	13.678	104	0.831331	8	0.527428	12
	2*	14.441	104	0.817334	9	0.519821	14
	4*	14.581	104	0.814229	10	0.518513	15
STA-F-83	21	160.746	227	0.888536	1	0.215426	1
	20	161.151	227	0.887829	2	0.214107	2
	15	164.375	228	0.871792	3	0.202156	3
	3	167.394	227	0.824391	4	0.196779	4
	31	168.195	227	0.805614	5	0.194967	5
	18	168.863	227	0.788882	6	0.193535	6
	11*	168.781	232	0.788385	7	0.182500	17
	16*	169.100	227	0.782864	8	0.193043	7
	29*	171.249	227	0.733062	9	0.188900	8
	9*	171.391	227	0.730410	10	0.188645	9
TRE-S-92	19*	9.311	69	0.880078	1	0.478231	2
	8*	9.389	68	0.878204	2	0.479078	1
	20	9.598	68	0.871588	3	0.475325	3
	7*	9.039	75	0.868946	4	0.468005	6
	6*	9.757	71	0.864316	5	0.465758	8
	17*	9.885	68	0.858365	6	0.469941	4
	21*	8.671	77	0.855435	7	0.469016	5
	1*	10.003	68	0.851293	8	0.467596	7
	10	9.856	75	0.846708	9	0.454514	9
	16*	9.981	77	0.826007	10	0.446743	11
UTA-S-92	17	3.567	63	0.888536	1	0.532771	1
	11	3.833	68	0.878185	2	0.511100	2
	14	3.911	68	0.876019	3	0.508369	3
	13	3.927	68	0.875482	4	0.507798	4
	16	3.977	68	0.873738	5	0.506065	5
	12	4.143	68	0.866816	6	0.500466	6
	24	4.531	73	0.807693	7	0.475697	7
	23	4.573	73	0.802872	8	0.474319	8
	27	4.581	73	0.801938	9	0.474053	9
	8	4.976	68	0.762605	10	0.472232	10
UTE-S-92	19	30.323	83	0.879116	1	0.438284	1
	18	29.718	86	0.878651	2	0.429775	2
	21	28.069	90	0.853031	3	0.420748	3
	20	32.804	88	0.835146	4	0.400981	4
	26	31.522	91	0.826953	5	0.392480	5
	15	33.935	91	0.780095	6	0.378000	6
	27	34.928	90	0.767341	7	0.377994	7
	12*	32.996	94	0.758297	8	0.367082	9
	17*	29.695	98	0.723270	9	0.369027	8
	8	30.555	98	0.721926	10	0.362837	10
YOR-F-83	21	39.801	234	0.883004	1	0.372139	1
	8*	44.158	233	0.837983	2	0.363036	3
	20*	44.412	231	0.831362	3	0.365581	2
	9	45.645	228	0.791749	4	0.359602	4
	14	45.736	238	0.785008	5	0.345675	5
	1	46.810	234	0.751639	6	0.341781	6
	2	46.862	235	0.749650	7	0.340088	7
	17	47.142	240	0.736830	8	0.330597	8
	32*	46.947	244	0.731929	9	0.324728	10
	31*	47.396	242	0.726141	10	0.324908	9

4 Discussion

The fuzzy system presented here provides a mechanism to allow an overall decision in evaluating the quality of a timetable solution to be made based on common sense rules that encapsulate the notion that the timetable solution quality increases as both the *average penalty* and the *highest penalty* decrease. The rules are in a form that is easily understandable by any timetabling officer.

Looking at Figures 4(a) and 4(b) it can be seen that, in many cases, it is not guaranteed that timetable solutions with low *average penalty* will also have low *highest penalty*. This observation confirmed the assumption that considering only the proximity cost to measure timetable solution quality is not sufficient. As an example, if the detailed results obtained for the $[0, maxValue]$ boundary range for *LSE-F-91* in Table 3 are analysed, it can be seen that solution 13 (with the lowest *average penalty*) is not ranked as the 'best' solution. The same effect can be observed in solution 21 for the *TRE-S-92* data set and solution 21 for the *UTE-S-92* data set in Table 4.

In these three data sets (*LSE-F-91*, *TRE-S-92* and *UTE-S-92*), the timetable solutions with the lowest *average penalty* were not selected as the 'best' timetable solution, because the decision made by the fuzzy evaluation system also takes into account another criterion, the *highest penalty*. This finding can also be seen in the other data sets, but it is not too obvious especially if we only focus on the first 3 'best' solution. Regardless, in terms of functionality, these results indicate that the fuzzy evaluation system has performed as intended in measuring the timetable's quality by considering two criteria simultaneously.

Analysing Tables 3 and 4 further, it can also be observed that the decision made by the fuzzy evaluation function is affected slightly when the different boundary settings are used to normalise the input values. The consequence of this is that the same timetable solution might be ranked in a different order, dependent on the boundary conditions. In both tables, the solutions with different ranking position are marked with *. For the *CAR-F-92* (in Table 3) and *UTA-S-92* data sets (in Table 4), the solution rankings are unchanged by altering the boundary settings. In several cases, the solution rankings are only changed slightly. It is also interesting to note that, in a few cases, for example solution 3 for *KFU-S-93* (in Table 3) and solution 11 for *STA-F-83* (in Table 4), the ranking change is quite marked.

Overall, the performance of the fuzzy evaluation system utilizing the boundary range $[0.0, maxValue]$ did not seem as satisfactory as when the boundary range $[minValue, maxValue]$ was used. This observation is highlighted by Table 5, which presents the fuzzy quality measure obtained for the 'worst' and 'best' solutions as evaluated under the two different boundary settings.

When the boundary range $[0.0, maxValue]$ was used, it can be seen that the fuzzy evaluation system evaluated the quality of the timetable solutions for the 12 data sets in the overall range of 0.111464 to 0.610225. In the case of *STA-F-83*, the 'best' solution was only rated as 0.215426 in quality. The quality of timetable solutions falls only in the regions of linguistic terms that correspond to meanings of *very low*, *low* and *medium* in the timetable *quality* fuzzy set (see Figure 2(c)).

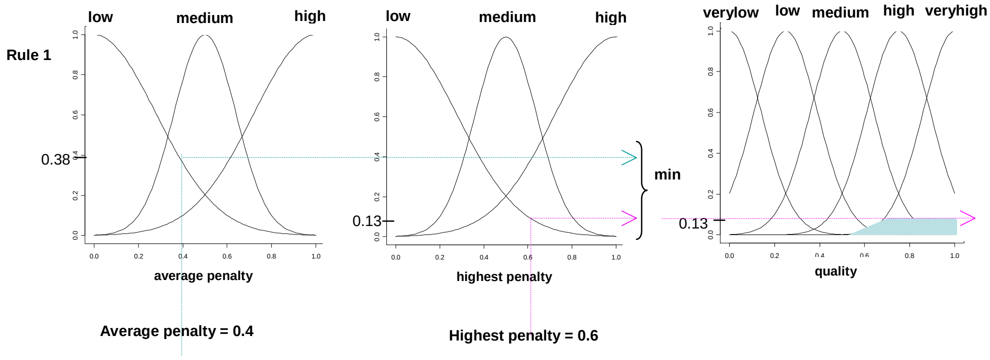
Table 5. Range of timetable quality

Data Set	Range $[0, maxValue]$		Range $[minValue, maxValue]$	
	Worst Solution	Best Solution	Worst Solution	Best Solution
CAR-F-92	0.111464	0.534427	0.111464	0.888503
CAR-S-91	0.111464	0.557585	0.111464	0.888524
EAR-F-83	0.111465	0.467867	0.111465	0.868135
HEC-S-92	0.127502	0.506506	0.155374	0.863057
KFU-S-93	0.111466	0.541211	0.111466	0.888529
LSE-F-91	0.111895	0.555747	0.112182	0.881499
RYE-F-92	0.115999	0.610225	0.119240	0.888528
STA-F-83	0.111464	0.215426	0.111464	0.888536
TRE-S-92	0.111476	0.479078	0.111488	0.880078
UTA-S-92	0.111464	0.532771	0.111464	0.888536
UTE-S-92	0.111464	0.438284	0.111464	0.879116
YOR-F-83	0.120046	0.372139	0.213388	0.883004

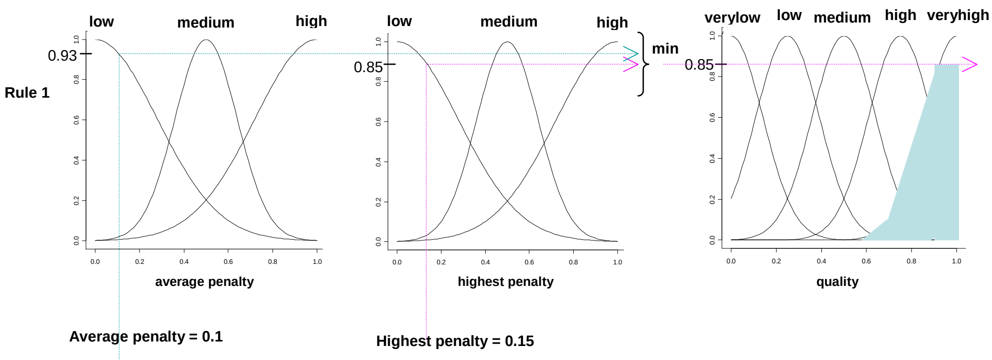
This is because the lower bound value used here (i.e. $lowerBound = 0.0$) is far smaller than the actual smallest values. Consequently, the input values for even the lowest values (i.e. the 'best' solution qualities) are transformed to normalized values that always fall within the regions of the *medium* and *high* linguistic terms in the input variables. As a result, the normalized input values will not cause any rule to be fired or, the firing level for any rule is relatively very low. This is illustrated in Figure 5(a), in which the activation level of the consequent part for **Rule 1** is equal to 0.13. Although the possibility exists for any input to fall into more than one fuzzy set, so that more than one rule can be fired, the aggregation of fuzzy output for all rules will obtain a final shape that will only produce a low defuzzification value.

In contrast, Figure 5(b) illustrates the situation when the normalised input values fall in the regions of linguistic term that corresponding to the meaning of *low*. In this situation, a high defuzzification value will be obtained due to the fact that most of the rules will have a high firing level. Thus, all of the solutions being ranked first had quality values more than 0.8, when the initial range $[minValue, maxValue]$ was used. In this case, the quality of timetable solutions falls in the regions of the linguistic terms that correspond to meanings of *high* and *very high* for the timetable *quality* fuzzy set (see Figure 2(c)). As might be expected, from the fact that the actual minimum and maximum values from the 35 constructed timetable solutions were used, the fuzzy evaluation results were nicely distributed along the universe of discourse of the timetable *quality* fuzzy set.

For a clearer comparison of the effect of the two boundary settings, the distribution of input and output values for the *UTA-S-92* data set are presented in Figure 6. As can be seen, the input values (Figures (b) and (c))



(a) Normalised value falls in the middle regions of the universe of discourse



(b) Normalised value falls in the left regions of the universe of discourse

Fig. 5. Firing level for **Rule 1** with different normalized input values

are concentrated in the middle regions $(0.4 - 0.7)$ of the graphs when the boundary range $[0.0, maxValue]$ was used. In contrast, when the boundary range $[minValue, maxValue]$ was used, the input values were concentrated in the bottom regions of the graphs. Based upon the defined fuzzy rules, we know that the timetable quality increases with a decrease in both input values. Indeed, this behavior of the output can be observed for both boundary setting (see Figure 6(a)). Using either of the boundary settings, the fuzzy evaluation system is capable of ranking the timetable solutions. It is purely a matter of choosing the appropriate boundary settings of the fuzzy sets for the input variables.

One of the deficiencies of this fuzzy evaluation, at present, appears to be that there is no simple way of selecting the boundary settings of the input variables. The drawback is that both boundary settings implemented so far can only be applied after a number of timetable solutions are generated. Therefore significant amounts of times are required to construct and analyse the solutions. Furthermore, if boundary setting are based on the actual minimum and maximum values

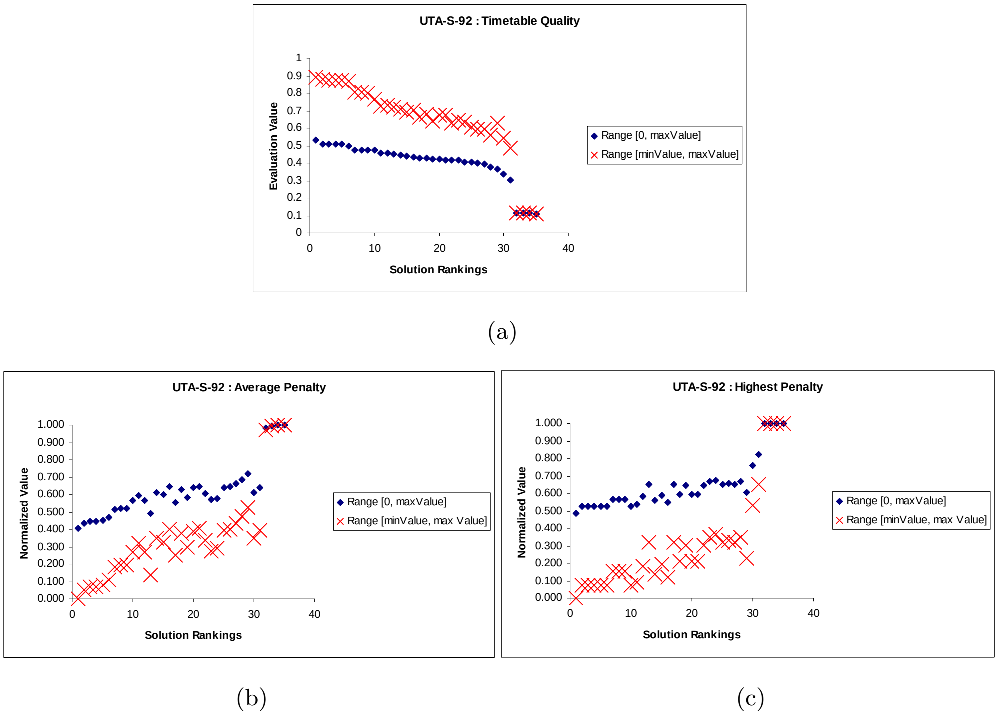


Fig. 6. A graphical comparison of the effect of the two boundary settings for UTA-S-92

from the existing timetable solutions, the fuzzy evaluation system might not be able to evaluate a newly constructed timetable solution if the input values for the decision criteria for the new solution lie outside the range of the fuzzy sets. (Actually, output values *can* always be calculated — the real problem is that the resultant solution quality will always be the same once both criteria reach the left-hand boundary of their variables.) Thus it would be highly beneficial if we could determine approximate boundary settings, particularly some form of estimate of the lower bound of the assessment criteria, based upon the problem structure itself.

5 Conclusions

In conclusion, the experimental results presented here demonstrate the capability of a fuzzy approach of combining multiple decision criteria in evaluating the overall quality of a constructed timetable solution. However, in the fuzzy system implementation the selection of the *lowerBound* and *upperBound* for the normalisation process is extremely important because it has a significant effect on the overall quality obtained. The initial results presented here only use two decision criteria to evaluate the timetable quality. Possible directions for future research include extending the application of the fuzzy evaluation system by considering more criteria, and devising a more sophisticated approach to determine approximate boundary settings for the normalisation process. Another aspect to be investigated further is in comparing the quality assessments produced by such fuzzy approaches with the subjective assessments of quality that timetabling officers make in real-world timetabling problems.

Acknowledgements. This research work is supported by the Universiti Teknologi Malaysia (UTM) and the Ministry of Science, Technology and Innovation (MOSTI) Malaysia.

References

1. S. Abdennadher and M. Marte. University Course Timetabling Using Constraint Handling Rules. *Journal of Applied Artificial Intelligence*, 14(4):311–326, 2000.
2. H. Asmuni, E. K. Burke, and J. M. Garibaldi. A Comparison of Fuzzy and Non-Fuzzy Ordering Heuristics for Examination Timetabling. In A. Lotfi, editor, *Proceeding of 5th International Conference on Recent Advances in Soft Computing 2004*, pages 288–293, 2004.
3. H. Asmuni, E. K. Burke, J. M. Garibaldi, and Barry McCollum. Fuzzy Multiple Heurisitc Orderings for Examination Timetabling. In Burke and Trick [13], pages 334–353.
4. P. Boizumault, Y. Delon, and L. Peridy. Constraint Logic Programming for Examination Timetabling. *The Journal of Logic Programming*, 26(2):217–233, 1996.
5. E. K. Burke, Y. Bykov, J. Newall, and S. Petrovic. A Time-Predefined Local Search Approach to Exam Timetabling Problems. *IIE Transactions*, 36(6):509–528, June 2004.
6. E. K. Burke, D. G. Elliman, P. H. Ford, and R. F. Weare. Examination Timetabling in British Universities - A Survey. In Burke and Ross [12], pages 76–90.
7. E. K. Burke, D. G. Elliman, and R. F. Weare. A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems. In *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95, Pittsburgh, USA, 15th-19th July 1995)*, pages 605–610, San Francisco, CA, USA, 1995. Morgan Kaufmann.
8. E. K. Burke, G. Kendall, and E. Soubeiga. A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics*, 9(6):451–470, Dec 2003.
9. 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.
10. E. K. Burke, J. P. Newall, and R. F. Weare. A Memetic Algorithm for University Exam Timetabling. In Burke and Ross [12], pages 241–250.
11. E. K. Burke, S. Petrovic, and R. Qu. Case Based Heuristic Selection for Timetabling Problems. *Journal of Scheduling*, 9(2):99–113, 2006.
12. Edmund K. Burke and Peter Ross, editors. *Practice and Theory of Automated Timetabling, First International Conference, Edinburgh, U.K., August 29 - September 1, 1995, Selected Papers*, volume 1153 of *Lecture Notes in Computer Science*. Springer, 1996.
13. Edmund K. Burke and Michael A. Trick, editors. *Practice and Theory of Automated Timetabling V, 5th International Conference, PATAT 2004, Pittsburgh, PA, USA, August 18-20, 2004, Revised Selected Papers*, volume 3616 of *Lecture Notes in Computer Science*. Springer, 2005.

14. M. W. Carter, G. G. Laporte, and S. Y. Lee. Examination Timetabling: Algorithmic Strategies and Applications. *Journal of the Operational Research Society*, 47:373–383, 1996.
15. S. Casey and J. Thompson. GRASPing the Examination Scheduling Problem. In E. K. Burke and P. D. Causmaecker, editors, *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers)*, volume 2740 of *Lecture Notes in Computer Science*, pages 232–244, Berlin Heidelberg New York, 2003. Springer-Verlag.
16. E. Cox and M. O'Hagen. *The Fuzzy Systems Handbook : A Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems* . AP Professional, Cambridge, MA, 1998.
17. S. Deris, S. Omatu, H. Ohta, and P. Saad. Incorporating Constraint Propagation in Genetic Algorithm for University Timetabling Planning. *Engineering Applications of Artificial Intelligence*, 12:241–253, 1999.
18. L. Di Gaspero and A. Schaerf. Tabu Search Techniques for Examination Timetabling. In E.K. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling III (PATAT 2000, Konstanz Germany, August, selected papers)*, volume 2079 of *Lecture Notes in Computer Science*, pages 104–117, Berlin Heidelberg New York, 2001. Springer-Verlag.
19. Christelle Guéret, Narendra Jussien, Patrice Boizumault, and Christian Prins. Building University Timetables Using Constraint Logic Programming. In Burke and Ross [12], pages 130–145.
20. Graham Kendall and Naimah Mohd Hussin. A Tabu Search Hyper-heuristic Approach to the Examination Timetabling Problem at the MARA University of Technology. In Burke and Trick [13], pages 270–293.
21. Mamdani, E. H., and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.
22. S. Petrovic, V. Patel, and Y. Yang. University Timetabling With Fuzzy Constraints. In Burke and Trick [13].
23. R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.
24. J. M. Thompson and K. A. Dowsland. A Robust Simulated Annealing Based Examination Timetabling System. *Computers & Operations Research*, 25(7/8):637–648, 1998.
25. Hiroaki Ueda, Daisuke Ouchi, Kenichi Takahashi, and Tetsuhiro Miyahara. Comparisons of Genetic Algorithms for Timetabling Problems. *Systems and Computers in Japan*, 35(7):1–12, 2004. Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol.J86-D-I, No. 9, September 2003, pp. 691-701.
26. G. M. White, B. S. Xie, and S. Zonjic. Using Tabu Search with Longer-Term Memory and Relaxation to Create Examination Timetables. *European Journal of Operational Research*, 153:80–91, 2004.
27. Y. Yang and S. Petrovic. A Novel Similarity Measure for Heuristic Selection in Examination Timetabling. In Burke and Trick [13].
28. L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.
29. H. J. Zimmerman. *Fuzzy Set Theory and Its Applications.* Kluwer Academic Publishers, 3rd edition, 1996.