Int. J. Fuzzy Syst. (2016) 18(1):119–129
DOI 10.1007/s40815-015-0046-z

A Fuzzy Logic and Binary-Goal Programming-Based Approach for Solving the Exam Timetabling Problem to Create a Balanced-Exam Schedule

Fatih Cavdur[1] · **Merve Kose**[1]

Received: 13 November 2014 / Revised: 23 April 2015 / Accepted: 6 May 2015 / Published online: 29 May 2015
© Taiwan Fuzzy Systems Association and Springer-Verlag Berlin Heidelberg 2015

Abstract This study presents a fuzzy logic and binary-goal programming-based approach for solving the exam timetabling problem to create a *balanced-exam schedule*. To be able to address the practical challenges of the exam timetabling problem, the model is developed with and verified by a human expert for exam scheduling. We propose a fuzzy-criticality level identification methodology to assign the criticality levels of exams for the students using three pieces of information, namely, credits, success ratios, and types of the classes. It is noted that the computed criticality levels are close approximates for those of the human expert. We then present a goal programming model to schedule exams using these criticality levels as well as other general problem data. The result of the goal program is a balanced-exam schedule in terms of exam criticality levels. Final step includes room assignments using a simple algorithm. The significance of the study is the consideration of the exam criticalities, for not only the students of the same year but also the students with different levels of seniority, as well as an even distribution of exams for professors which make the problem more challenging for the human expert in practice. Using a real-life problem, we show that our approach creates an exam schedule that is more preferable than the one prepared by the human expert. Additionally, computational results show the potential of our model to be used in real-life problems of larger-size.

Keywords Exam timetabling · Balanced-exam schedule · Integer programming · Goal programming · Multi-criteria optimization · Fuzzy logic

1 Introduction

Exam scheduling, often referred as the *exam timetabling problem*, is a challenging real-life problem since it not only includes general problem restrictions, but also involves the preferences or requests of several groups of people, such as students, professors, proctors, etc. In some cases, rather than the general problem restrictions, it is the preferences or requests of those groups that make the problem more challenging since the decision maker(s) responsible for the exam schedule (referred as the human expert(s) from now on) might have much more difficulty in satisfying—probably the conflicting—criteria corresponding to such preferences or requests. As a result, since it is usually not possible to find a solution perfectly suited to each group, the human expert in general follows a practical solution approach for solving the exam timetabling problem for creating an exam schedule, that is aimed at being at least relatively fair to each of those groups, what we refer as a *balanced-exam schedule* in this study.

What motivates this study is the increasing number of complaints, especially from the students, related to the exam schedules in the Department of Industrial Engineering of Uludag University. In the Department of Industrial Engineering, there are two exam periods in each semester, one for the midterms and one for the finals. In each exam period, students usually have 9–10 exams which is the case for a regular student. The number of exams in each exam period is even more for those students with extra coursework (who had failed some classes or who are registered

✉ Fatih Cavdur
fatihcavdur@uludag.edu.tr

[1] Department of Industrial Engineering, Uludag University, Nilufer, Bursa, Turkey

for upper-level courses). Naturally, the students are concerned about when their exams are scheduled during an exam period, and thus, most of the complaints are about how their exams are distributed during the exam period. The students expect to have an even exam distribution, in terms of not only the numbers but also the *criticalities* of exams. In other words, the workloads of the exams of different classes might also differ significantly. For instance, in a particular exam day, the students might prefer having three exams with low-criticality to two exams with high-criticality. Consideration of the number of exams only without their criticalities might produce unsatisfactory results for the students. It thus becomes a very challenging task for the human expert to schedule exams considering the workloads of all students, especially since some students take classes of more than one semester. Moreover, in addition to the students, professors and proctors are also concerned about the distributions of their exams, and might also complain about the exam schedules. As a result, for the human expert, sometimes it takes a few weeks to create an exam schedule which is relatively acceptable by everyone.

We can define a balanced-exam schedule as the one that aims at balancing the schedule in terms of the workloads of the corresponding groups. In this study, we present a two-step solution approach to the problem based on the practical solution approach of one of the human experts responsible for the exam schedules in the Department of Industrial Engineering. Our objective is to represent all the details of the reasoning of the human expert in our model so that it can be used to solve the real-life problems by the human experts. In the first phase of our approach, *criticality levels* of exams are identified using a fuzzy-criticality level identification methodology considering the credits, success ratios, and types (core or elective) of the classes as well as the reasoning of the human expert for deciding the criticality levels of the exams. This step, in fact, simulates the decision-making process of the human expert to identify the criticality levels of the exams. The human expert knows, for instance, if a core class has a low success ratio, then, this class' exam is very critical for the students. Using the same reasoning, we design our fuzzy-criticality level identification methodology, details of which explained in the following pages. The second step of the approach is then uses the criticality level information from the first phase as well as some other pre-processed problem data as its input and creates an optimal exam schedule using a goal programming model as seen in Fig. 1. Finally, using this balanced-exam schedule, we perform room and proctor assignments using a simple algorithm.

The paper is organized as follows. The next section summarizes the related studies in the literature. Section 3 explains the methodology and the details of the fuzzy logic and goal programming-based model, and presents a simple algorithm to perform room and proctor assignments. In Sect. 4, we present the solution of our model as well as a comparison of the exam schedules of the model and the human expert. Finally, conclusion remarks are summarized in Sect. 5.

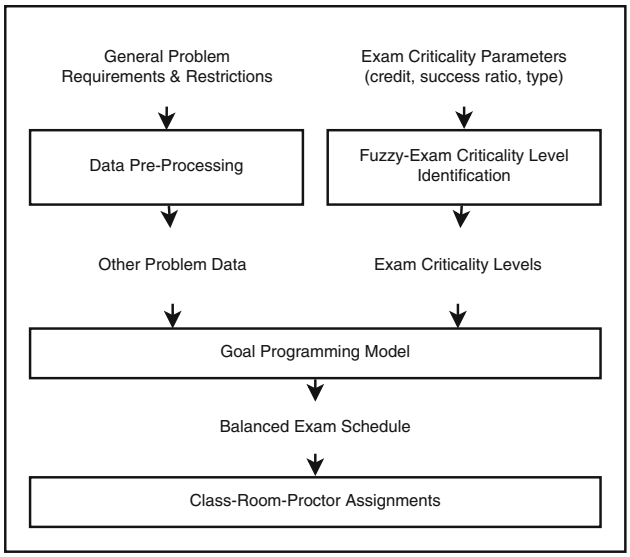


Fig. 1 Overall solution approach

2 Background and Related Studies

The timetabling problem is among the well-studied problems in operations research [1]. A classical example of the timetabling problem involves assigning classes or exams to time slots and/or rooms optimally. Such a problem might include various types of constraints depending on the complexity of the problem. As a well-known operations research problem, although a vast literature exists on the timetabling problem, it is still difficult to find a model that fits-all, since the problem instances significantly varies for different institutions, and even for the sub-divisions of the same institution such as the faculties or departments in a university. Comprehensive formulations for some variations of the timetabling problem are developed in previous studies. Al-Yakoob and Sherali [2] consider a various number of constraints in their study in addition to general problem restrictions including the dissatisfaction of faculty members and gender-based policies. Another comprehensive formulation is presented by Daskalaki et al. [3] where the authors consider teaching preferences, compactness of student schedules, and minimization of room changeovers in addition to the hard constraints of the university timetabling problem. Another study by Daskalaki and Birbas [4] proposes a relaxation approach to solve the integer

programing model for a university timetabling problem as the computational complexity is an important issue. Another solution approach is dividing the problem into two phases in which time and room assignments are performed, respectively [5]. Al-Yakoob and Sherali [6] present a column generation approach to solve the high-school timetabling problem in a recent study. Different approaches to solve the exam timetabling version of the timetabling problem are also presented, such as heuristic ordering-based methods and meta-heuristics [7–9]. In [10, 11], and [12], surveys of university timetabling approaches are presented. For general reviews on timetabling problems, the reader may refer to Schaerf [13] and Schmidt and Strohlein [14].

The exam timetabling problem is one of the specializations of the general timetabling problem. Just as the general timetabling problem, the exam timetabling problem is also considered in many studies. Several comprehensive reviews are presented in [15–18]. Some studies involve the development of a software package to perform exam timetabling, such as EXAMINE by Carter et al. [19]. In the formulation of the exam timetabling problem, there are mainly two types of constraints considered in previous studies. The first type of constraints, namely, the hard constraints, is mostly used to specify some problem restrictions, such as not scheduling more than one exam to a room during the same time slot. These hard constraints, on the other hand, are usually not sufficient to create an optimal exam schedule in terms of satisfying the requests or necessities of students, professors, research assistants, proctors, etc. where we employ the constraints of the second type, the soft constraints. In a real-life exam timetabling problem, consideration of the availability/unavailability and pre-assignments might also be necessary. An exam timetabling problem might also include room assignments. In such cases, the problem can be further classified as a single room problem or a multiple-room problem in which a single room or more than one room can be used for a single exam, respectively. Conversely, in some cases, scheduling more than one exam to a single room during the same time slot is allowed. A comprehensive mathematical model including a variety of problem specifications is presented in [20] where the authors partitioned the problem into two sub-problems where (i) time and room assignments and (ii) proctor assignments are performed, respectively.

It is usually not possible to find an ideal solution to the exam timetabling problem that satisfies both hard and soft constraints at the same time. In other words, an ideal solution that satisfies all criteria of the problem does not exist in general [21]. Sometimes it might be desirable to model the exam timetabling as a constraint satisfaction problem, especially if quite many of such criteria exist [22]. Under the circumstances, to create a good (or at least an acceptable) exam schedule, the solution to the problem must be feasible (i.e., must satisfy the hard constraints) and should be ideal at some degree (i.e., should satisfy the soft constraints as much as possible). The definition of a good solution, however, might differ depending on the institution where the problem applies. In some institutions, for instance, it is required or at least desired to have one-day-off between two consecutive exams. On the other hand, it might not be possible to achieve that goal due to time restrictions in many cases. Nevertheless, with some variations, it is still one of the crucial issues for creating exam schedules in most institutions; that is, scheduling exams so that the workloads of students, professors, research assistants, proctors, etc. are evenly distributed during the exam period. In other words, creating a fair schedule for all groups of interest (i.e., students, professors, research assistants, proctors, etc.), a *balanced-exam schedule* as we refer in this study, is very important. Since it is usually not possible to satisfy the requests or necessities of all of these groups of interest at the same time, especially in larger institutions where several restrictions apply, we note that the human experts responsible for the exam schedules follow some practical guidelines to create a balanced-exam schedule spending a considerable amount of time. The model presented in this study follows the reasoning of a human expert in an institution to create an optimal exam schedule (i.e., a perfectly balanced-exam schedule).

Since satisfying the requests or necessities of several groups of people is one of our concerns, the exam timetabling problem is usually formulated as a multi-criteria optimization problem. Among other approaches, goal programming models, as in this study, are extensively used in previous studies for solving the exam timetabling problem. Typical goals in such a goal programming model are generally related to distributing exams evenly for students, professors, research assistants, proctors, etc., or more specifically, it is desired to maximize the study time for each student, for instance [23]. In particular, these involve setting lower–upper bounds for the number of exams and minimizing or maximizing the number of exams during a particular time period, such as per day, per week, etc. to create a balanced-exam schedule as in our model presented in this study [24]. One of the challenges for creating a balanced-exam schedule, which makes the problem more realistic and complicated at the same time, is the consideration of intangible or qualitative factors such as satisfaction degree for a group of people. Among the most important criteria affecting the degree of satisfaction, especially for students, is the criticality levels of exams, as we refer in this study, since the workloads of students (and in some cases, also of professors, research assistants, proctors, etc.) might significantly differ for the exams of

different classes depending on how critical the exams are for the students. An approach to overcome the difficulty of identifying intangible or qualitative factors, criticality levels for instance, might include a procedure, such as the analytic hierarchy process (AHP) or the analytic network process (ANP), for the identification of such intangible or qualitative factors and input the results of the procedure to the multi-objective optimization model (i.e., the goal program) [25, 26]. Since one of the most important criteria affecting the degree of satisfaction, especially for students, is the criticality levels of exams, as we refer in this study, because of the significantly varying workloads of students (and in some cases, also of professors, research assistants, proctors, etc.) for the exams of different classes depending on how critical the exams are for the students; an alternative way of quantifying such intangible or qualitative factors is using these criticality levels. In practice, identification of these *criticality levels*, on the other hand, might require following the reasoning of the human expert(s) responsible for exam scheduling in the corresponding institutions, where we might consider utilizing artificial intelligence-based approaches, such as the fuzzy set theory. Since its introduction by Zadeh in 1965 [27], fuzzy approaches are applied in various fields [28]. Some example reviews on these applications can be found in [29–32]. We note that fuzzy approaches are frequently used in decision making to model the uncertainties or fuzziness in the decision making process [33–36]. Interesting educational applications are also presented. Tsai and Yeh, for instance, propose a fuzzy logic approach to classify student's attentiveness [37]. According to the best of our knowledge, however, although there exist a few studies including fuzziness for solving the exam timetabling problem [38], integrating these fuzzy approaches by considering the fuzziness of intangible or qualitative factors to a the multi-objective optimization model to create a balanced-exam schedule has not yet been suggested; motivating the approach presented in this study.

3 Methodology

3.1 Fuzzy-Criticality Level Identification

This section explains the details of the fuzzy-criticality level identification methodology. We design a 3-input, 1-output, 12-rule fuzzy system to identify the criticality levels of the exams. The reasoning of the human expert is used in the design of the system. We note that the human expert basically considers three parameters to assign the exams to the time slots which are the (i) credits, (ii) types (core or elective), and (iii) success ratios (the ratio of the number of successful students to the number of all students) of the classes. The human expert is particularly concerned about the success ratios of the classes since the the exams of the classes with lower success ratios are considered more critical by the students of that class. Although the human expert can approximately identify the success ratios of the classes informally due to close student relationships, to use this parameter as an input for our model, we obtain the success ratios of all classes using the previous year's grade reports for each class. By considering these three criteria, the human expert identifies, as we refer in this study, the *criticality levels* of the exams. To follow the same reasoning as the human expert, we define the inputs as the credits, success ratios, and types of the classes, and the only output as the criticality levels.

We use Gaussian type membership functions as in Eq. 1 for the inputs class credits and success ratios with parameters μ and σ, sigmoid-type membership function as in Eq. 2 for the input class type with parameters a and c, and a triangular-type membership function for the output with parameters a, b and c as given in Eq. 3, respectively. Since the first two inputs are more likely to be represented using several levels, such as low, average, high, etc., a Gaussian type membership functions is a typical choice to model each level of each input with fuzziness. On the other hand, as the third input corresponds to a binary quantity (a class is either core or elective), two sigmoid-type membership functions are used to represent a core and elective class. For the output, we use five triangular-type membership functions to simply represent the five different levels of criticality. The reason that we assume Gaussian type membership functions for the first two inputs and a triangular-type membership function for the output is that we want to emphasize the fuzziness for the class credits and success ratios using a nonlinear function (Gaussian) and simplify the output using a linear function triangular. Nevertheless, it is possible to use them interchangeably or simply use the same type of membership functions for both inputs and the output.

$$f(x;\mu,\sigma) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

$$f(x;a,c) = \frac{1}{1+e^{-a(x-c)}} \tag{2}$$

$$f(x;a,b,c) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & x > c \end{cases} \tag{3}$$

We assume three different types of classes in terms of their credits, as low-credit, average-credit, and high-credit classes, and thus, set the parameters of the membership function of input 1 as $(0.375, 1.5)$, $(0.375, 2.5)$ and

$(0.375, 3.5)$, respectively. Similarly, for success ratios, we define three different levels as low, average, and high as well, and let the corresponding membership function parameters be $(0.1, 0.1)$, $(0.1, 0.50)$ and $(0.1, 0.9)$. Since classes are classified either core or elective, we define two membership functions for input 3 with respective parameters $(-25, 0.5)$ and $(25, 0.5)$. Note that, with given parameters, each membership function corresponds to a different type of class in terms, of their credits (first membership function), success ratios (second membership function), and types (third membership function).

Finally, for the output, we assume five different criticality levels for the classes, as very low, low, average, high and very high, with the set of parameters $(0, 1, 2)$, $(1, 2, 3)$, $(2, 3, 4)$, $(3, 4, 5)$ and $(4, 5, 6)$, respectively. As for the inputs, each set of these parameters represents a class with a different level of criticality from very low (a criticality value of 1) to very high (a criticality value of 5). Table 1 presents a description of the membership functions for the inputs and the output where the inputs and the output are abbreviated as class credit (input 1, CC), class type (input 2, CT), success ratio (input 3, SR), and exam criticality level (output, ECL), respectively. The human expert identifies the criticality levels of exams by considering the three inputs. For instance, in general, the exam of a higher-credit, of a lower success ratio or of a core class is generally considered as more critical by the human expert. Our discussions with the human expert yielded the following rules.

1. `if CT is ELECTIVE and SR is LOW then ECL is AVG`
2. `if CT is ELECTIVE and SR is AVG then ECL is LOW`
3. `if CT is ELECTIVE and SR is HIGH then ECL is VERY LOW`
4. `if CC is LOW and CT is CORE and SR is LOW then ECL is AVG`
5. `if CC is AVG and CT is CORE and SR is LOW then ECL is HIGH`
6. `if CC is HIGH and CT is CORE and SR is LOW then ECL is VERY HIGH`
7. `if CC is LOW and CT is CORE and SR is AVG then ECL is LOW`
8. `if CC is AVG and CT is CORE and SR is AVG then ECL is AVG`
9. `if CC is HIGH and CT is CORE and SR is AVG then ECL is HIGH`
10. `if CC is LOW and CT is CORE and SR is AVG then ECL is VERY LOW`
11. `if CC is AVG and CT is CORE and SR is AVG then ECL is LOW`
12. `if CC is HIGH and CT is CORE and SR is AVG then ECL is AVG`

As a result, the mappings between the inputs and the output of the fuzzy-criticality level identification system are shown in Figs. 2, 3 and 4. These figures correspond to the mappings of class credit and class type versus exam criticality levels, class credit and success ratio versus exam criticality levels, and class type and success ratio versus exam criticality levels, respectively. We note how these mappings represent the human expert's identification of the criticality levels. For instance, in Fig. 2, it is noted that the criticality level of an exam increases by the class credit of a core class, whereas if the corresponding class is elective, criticality of the exam does not change by the class credit. Note that, it is just the criticality level identification of the human expert as represented by the first three rules. Figures 3 and 4 show how success ratio affects the criticality level of an exam, where, we note that, although the criticality

Table 1 Membership functions

Fuzzy I/Os	Type	Value	Parameters
Input (CC)	Gaussian	Low	$(0.375, 1.5)$
		Average	$(0.375, 2.5)$
		High	$(0.375, 3.5)$
Input (CT)	Sigmoid	Core (0)	$(-25, 0.5)$
		Elective (1)	$(-25, 0.5)$
Input (SR)	Gaussian	Low	$(0.1, 0.10)$
		Average	$(0.1, 0.50)$
		High	$(0.1, 0.90)$
Output (ECL)	Gaussian	Very low	$(0, 1, 2)$
		Low	$(1, 2, 3)$
		Average	$(2, 3, 4)$
		High	$(3, 4, 5)$
		Very high	$(4, 5, 6)$

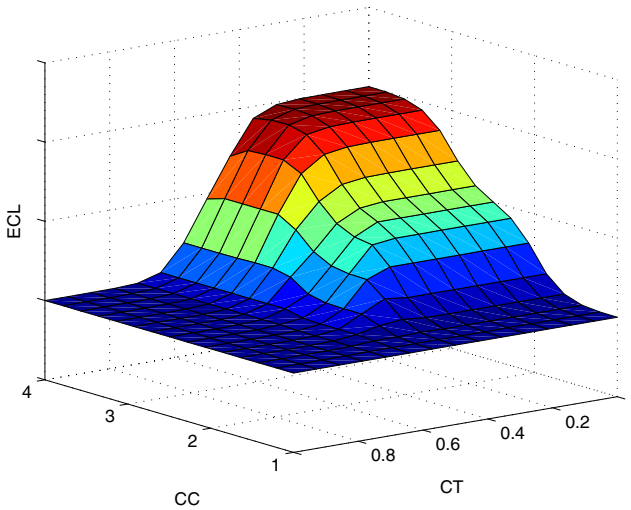


Fig. 2 I/O mapping—inputs: class credit and class type

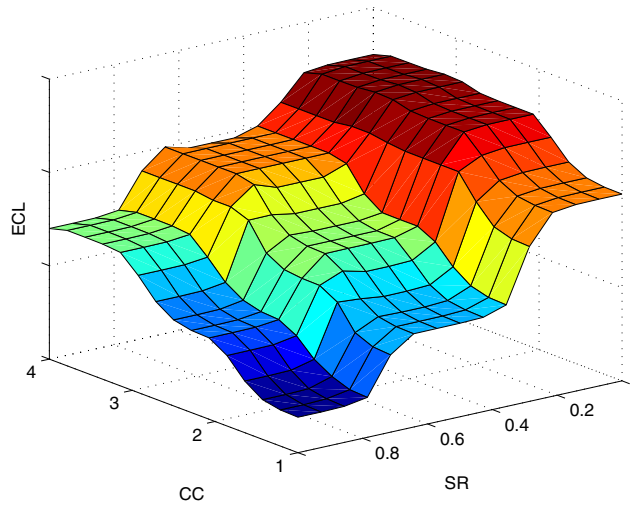


Fig. 3 I/O mapping—inputs: class credit and success ratio

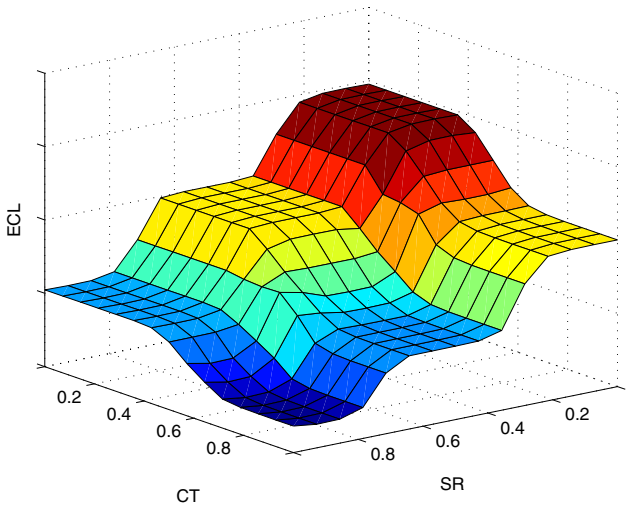


Fig. 4 I/O mapping—inputs: class type and success ratio

level increases for lower success ratios, the most critical exams are of those with the highest credits core classes. According to our approach, after identifying the criticality levels of the exams, the exams should be evenly distributed in terms of their criticality levels (subject to the other problem constraints) as much as possible, details of which are presented in the next section.

3.2 Goal Programming Model

Our goal programming model is presented in this section. We have a 2 week-workday exam period, that is, there are 10 days in the exam period, starting from the Monday of the first week until the Friday of the second week with Saturday and Sunday excluded in between. On each day, there are four time slots to schedule the exams as the first time slot is from 9.00 A.M. to 10.45 A.M., the second time slot is from 11.00 A.M. to 12.45 P.M., and so on. No exam lasts more than 1 h and 45 min, so that it does not occupy more than one time slot and allow the human expert schedule another exam for the next time slot. As the main concern of our study is to create a balanced-exam schedule, we are not focused on room and proctor assignments primarily, and thus, our goal program does not consider room and proctor assignments. However, for the sake of completeness, a simple algorithm is presented in the next subsection for performing room and proctor assignments. We have the following parameters for the goal program.

- i: class, $i = 1, \ldots, n_I$, $i \in I$, set of all classes.
- j: professor, $j = 1, \ldots, n_J$, $j \in J$, set of all professors.
- k: semesters, $k = 1, \ldots, n_K$, $k \in K$, set of all semesters.
- t: time slot, $t = 1, \ldots, n_T$, $t \in T$, set of all time slots.
- u: time slot (of each day), $u = 1, \ldots, 3$, $u \in t \in T$, set of all time slots.
- T_1, T_2: sets of time slots in the first and second weeks of the exam period, respectively.
- $T_F = 1, 5, 9, \ldots 37$: set of time slots that correspond to the first time slot of each day.
- $I_O = \{(i_1, i_2) \,|\, i_1, i_2 \in I, i_1 \neq i_2\}$: set of overlap-class-pairs (i.e., set of class-pairs that are taken by one or more students).
- c_i: criticality level of class i, computed using the fuzzy-criticality level identification methodology.
- c_{avg}: average criticality level per day, that is the total criticality level divided by the number of days in the exam period ($75/10 = 7.5$ per day in our case).
- $n_{\max}$: maximum number of exams per day, that is the number of time slots per day (4 in our case).

We let x_{ijkt} equal 1 if class i of professor j and semester k is assigned to time slot t and 0 otherwise. Following hard constraints are used to satisfy the basic requirements of an exam scheduling problem. We write the first set of constraints given by Eq. 4 to ensure that each class must be assigned to a time slot. The following set of constraints given by Eq. 5 ensures that there can be at most four exams in a day. Equation 6 is used to prevent assigning more than one class of the same semester to the same time slot. Finally by Eq. 7, the exams of the classes that are taken by the same students are assigned to different time slots (two classes cannot be assigned to the same time slot if one or more students take both classes).

$$\sum_{t=1}^{n_T} x_{ijkt} = 1, \quad \forall i \in I, j \in J, k \in K \tag{4}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{k=1}^{n_K}\sum_{u=0}^{3} x_{i,j,k,t+u} \le n_{\max}, \quad t \in T_F \tag{5}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J} x_{ijkt} \le 1, \quad \forall k \in K, \forall t \in T \tag{6}$$

$$\sum_{j=1}^{n_J}\sum_{k=1}^{n_K} x_{i_1,j,k,t} + \sum_{j=1}^{n_J}\sum_{k=1}^{n_K} x_{i_2,j,k,t} \le 1, \quad \forall t \in T, \forall (i_1, i_2) \in I_O \tag{7}$$

In addition to the requirements presented above, we have some criteria that are desired to be considered as much as possible. These are represented as the following goals.

(1) To schedule as many exams as the number of time slots (that is 4, in our case) in a day, each of which are from different semesters.
(2) To balance the workloads of the students in terms of the criticality levels of the classes.
(3) To evenly distribute the exams of the professors in both weeks of the exam period.

Equations 8 and 9 are used for the two parts of the first goal, respectively, that is, (i) to schedule as many exams as the number of time slots in a day, (ii) each of which are from different semesters.

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{k=1}^{n_K}\sum_{u=0}^{3} x_{i,j,k,t+u} \approx n_{\max}, \quad t \in T_F \tag{8}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{u=0}^{3} x_{i,j,k,t+u} \approx 1, \quad \forall k \in K, t \in T_F \tag{9}$$

The second goal is represented by Eqs. 10 and 11. The first one is used to balance the weekly semester-based workloads, whereas the second constraint aims at balancing the daily overall workloads. These two constraints serve for two different types of students, that is, Eq. 10 balances the workloads of the students of the same semester, whereas Eq. 11 balances the workloads of the students of different semesters. The reason for including the latter is due to the consideration of the students who take classes from different semesters (failed students, students taking upper-level classes, etc.) as suggested by the human expert. To balance the workloads of the students of different semesters using Eq. 11, a simple, and yet useful approach is to approximately have the same criticality level on each day, that is, the criticality levels of the exams on each day should approximately equal the average criticality level c_{avg}, which is $76/10 = 7.6$ per day in our case.

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{t\in T_1} c_i x_{ijkt} - \sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{t\in T_2} c_i x_{ijkt} \approx 0, \quad \forall k \in K \tag{10}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{k=1}^{n_K}\sum_{u=0}^{3} c_i x_{i,j,k,t+u} \approx \frac{\sum_{i=1}^{n_I} c_i}{n_I} = c_{\text{avg}}, \quad t \in T_F \tag{11}$$

Finally, to achieve the last goal, we use Eq. 12. This constraint evenly distributes the exams of the professors in 2 weeks as much as possible. In other words, for the professors who teach more than just one class, it is desired not to have all or most of their exams in either the first or the second week of the exam period, but rather have them on both weeks as even as possible. Just as the second goal, this constraint is also based on another suggestion of the human expert as the human is expected to balance the workloads of not only the students but also the professors.

$$\sum_{i=1}^{n_I}\sum_{k=1}^{n_K}\sum_{t\in T_1} x_{ijkt} - \sum_{i=1}^{n_I}\sum_{k=1}^{n_K}\sum_{t\in T_2} x_{ijkt} \approx 0, \quad \forall j \in J \tag{12}$$

By defining the corresponding deviational variables $d_t^{\pm 1}$, $d_{kt}^{\pm 2}$, $d_k^{\pm 3}$, $d_t^{\pm 4}$ and $d_j^{\pm 5}$ corresponding to the soft constraints, we can now write the followings.

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{k=1}^{n_K}\sum_{u=0}^{3} x_{i,j,k,t+u} - d_t^{+1} + d_t^{-1} = n_{\max}, \quad t \in T_F \tag{13}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{u=0}^{3} x_{i,j,k,t+u} - d_{kt}^{+2} + d_{kt}^{-2} = 1, \quad \forall k \in K, t \in T_F \tag{14}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{t\in T_1} c_i x_{ijkt} - \sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{t\in T_2} c_i x_{ijkt} - d_k^{+3} + d_k^{-3} = 0, \quad \forall k \in K \tag{15}$$

$$\sum_{i=1}^{n_I}\sum_{j=1}^{n_J}\sum_{k=1}^{n_K}\sum_{u=0}^{3} c_i x_{i,j,k,t+u} - d_t^{+4} + d_t^{-4} = \frac{\sum_{i=1}^{n_I} c_i}{n_I} = c_{\text{avg}}, \quad t \in T_F \tag{16}$$

$$\sum_{i=1}^{n_I}\sum_{k=1}^{n_K}\sum_{t\in T_1} x_{ijkt} - \sum_{i=1}^{n_I}\sum_{k=1}^{n_K}\sum_{t\in T_2} x_{ijkt} - d_j^{+5} + d_j^{-5} = 0, \quad \forall j \in J \tag{17}$$

For the sake of completeness, and also for being able to compare the result obtained from the model with the ones of the human expert, we add several other constraints

corresponding to the special requests of three professors. That is, the first professor wants to have her only exam scheduled at 9.00 A.M on the first Monday of the exam period. The second and third professors want to have their exams scheduled during only the first week and second week of the exam period, respectively. In addition, the third professor does not want to have her exams scheduled at 9.00 either. Although implemented in our operational (computer) model, these constraints are not presented here to keep the general structure of the model. Using the non-preemptive goal programming approach, and assuming that all goals are of equal importance by letting $w_1 = \cdots = w_5 = 1$, the objective function is written as follows.

$$\begin{aligned} \min z(\mathbf{x}, \mathbf{d}) = {} & w_1 \left(\sum_{t \in T_F} d_t^{-1} \right) \\ & + w_2 \left(\sum_{k=1}^{n_k} \sum_{t \in T_F} d_{kt}^{+2} + d_{kt}^{-2} \right) \\ & + w_3 \left(\sum_{t \in T_F} d_t^{+3} + d_t^{-3} \right) \\ & + w_4 \left(\sum_{t \in T_F} d_t^{+4} + d_t^{-4} \right) \\ & + w_5 \left(\sum_{j=1}^{n_j} d_j^{+5} + d_j^{-5} \right) \end{aligned} \tag{18}$$

3.3 Room and Proctor Assignments

Finally, although it is not crucial in our case and not the focus of our study, for the sake of completeness, after solving the goal programming model, class-room-proctor assignments are performed using Algorithm 1.

Algorithm 1 Class-Room-Proctor Assignment Algorithm

input: $C = \{(t_i, s_i) : \forall i\}$, $R = \{c_j : \forall j\}$ ▷ time, size of classes and capacity of rooms
output: $A = \{(i, j, k) : i \leftrightarrow j, i \leftrightarrow k, \exists i, j, k\}$ ▷ class-room-proctor assignments
local: C_U, R_U, P_U ▷ set of unassigned classes, rooms and proctors
$C_U \leftarrow \{i : \forall i\}$ ▷ initialize the set of unassigned classes
for all $t_i : \forall i$ **do**
  $R_U \leftarrow \{j : \forall j\}$ ▷ initialize the set of unassigned rooms
  $P_U \leftarrow \{k : \forall k\}$ ▷ initialize the set of unassigned proctors
  while $C_U \neq \emptyset \vee R_U \neq \emptyset \vee P_U \neq \emptyset$ **do**
    while $s_i > 0$ **do**
      $j \leftarrow j + 1$
      $s_i \leftarrow \max\{0, s_i - c_j\}$
      $R_U \leftarrow R_U \setminus \{j\}$
      for $u \leftarrow 1 : f(c_j)$ **do**
        $k \leftarrow k + 1$
        $P_U \leftarrow P_U \setminus \{k\}$
        $A \leftarrow A \cup \{(i, j, k)\}$
      end for
      if $s_i == 0$ **then**
        $i \leftarrow i + 1$
        $C_U \leftarrow C_U \setminus \{i\}$
      end if
    end while
  end while
end for

In Algorithm 1, the set of time and size pairs of classes C is in ascending order of time t_i for class i and the set of capacities of rooms R is in descending order of capacities c_j for room j. Note that Algorithm 1 assigns classes to rooms and proctors starting with the first class in the exam period and continues until all classes are assigned. We also note that, for each time slot t_i for class i, class-room-proctor assignments are performed until there are either no exams to assign or no rooms or proctors available at that particular time slot. The algorithm assumes that the required number of proctors as a function of room capacity ($f(c_j)$ for room j) are given. Resulting class-room assignments by Algorithm 1 are given with the exam schedule obtained by the proposed approach. The assignments performed by the human expert are also given in the corresponding exam schedule prepared by the human expert. As the proctors are assumed to be identical, proctor assignments are not shown in the exam schedules.

Table 2 Summary of results

Goal	Human expert	Model
1.(a)	4.0	2.0
1.(b)	4.0	3.0
2.(a)	9.0	1.0
2.(b)	23.0	5.0
3.(a)	30.0	22.0

4 Implementation and Results

We use the fuzzy-criticality level identification model to compute the criticality levels of the exams. The criticality levels computed by the model vary between 1 and 4 where 1 and 5 correspond to the least and most critical levels, respectively. In our problem, there are 17 exams with criticality levels of 1 all of which are of 2-or-less-credit elective classes whose average success ratio is 92 which are of high-credit-core classes with an average success ratio of 54 results shows that the computed criticality levels are close approximates for those of the human expert.

After computing the criticality levels of the exams, we solve the goal programming model assuming that each goal has an equal importance in 18. The exam schedule prepared by the human expert as well as the one obtained by the proposed approach along with the computed criticality levels of each exam are available upon request, and thus, are not presented in the paper due to space constraints. Results are summarized in Table 2 in terms of the goals in the previous section. A comparison of the schedules by the human expert and the model is as follows.

1. It is desired to schedule as many exams as the number of time slots (that is 4) in a day, each of which are from different semesters.
   (a) We see that, the human expert schedules 4 exams on 6 days, whereas the model schedules 4 exams on 8 days. In other words, the human expert and model are not able to schedule 4 exams on 4 days and 2 days, respectively.
   (b) We see that, the human expert does not schedule an exam from each semester on 4 days, whereas the model does not schedule an exam from each semester on 3 days.
2. It is desired to balance the workloads of the students in terms of the criticality levels of the classes.
   (a) We see that, the sum of the absolute values of the differences between the criticality levels of the exams in the first and second week are 9 and 1 for the human expert and the model, respectively.
   (b) We see that, the sum of the absolute values of the differences between the criticality levels of the exams in each day are 23 and 5 for the human expert and the model, respectively.
3. It is desired to evenly distribute the exams of the professors in both weeks of the exam period.
   (a) We see that, the sum of absolute values of the differences between the number of exams per week per professor are 30 and 22 for the human expert and the model, respectively.

We note that the schedule obtained using the model is better than the schedule prepared by the human expert in terms of all goals considered in the study. In addition, considering the time spent by the human expert to schedule the exams, our model is able to give the optimal solution in a few seconds totally, a significant result regarding the savings in terms of the human resources available.

5 Conclusions

This study presents a fuzzy logic and goal programming-based approach for the solution of the exam timetabling problem to create a balanced-exam schedule considering some of the practical challenges of the problem. Our approach is developed with a human expert responsible for exam scheduling in the Department of Industrial Engineering of Uludag University. In addition to the general problem restrictions, we also consider the exam criticalities for not only the students of the same year but also the students with different levels of seniority as well as an even distribution of the exams for the professors which make the problem more challenging for the human expert in practice. The results show that the exam schedule produced by our approach is more preferable than the one of the human expert in terms of satisfying the goals of the problem. Additionally, our approach is able to create an exam schedule for the problem instance considered in this study in a few seconds on a personal computer showing its potential to be used in real-life problems of larger-size.

The results of our approach are also important in which the exam schedule we present is useful in terms of preventing complains and conflicts among people since we create an exam schedule that is fair to everyone without asking everyone about their preferences, which, in most cases, is not possible in real-life. Utilization of our approach is also very significant in terms of the savings of the available human resources. For the human expert, it takes a few weeks to not only schedule the exams, but also reschedule them again and again due to the complaints and conflicts among people. Additionally, since it is usually not

very easy to understand the exam criticalities for the students for a human expert who does not have enough experience, developing approaches such as the one presented in this study to simulate the decision-making process of an experienced human expert is also very important. Finally, the flexibility of our approach allows it to be utilized in different institutions after appropriate modifications of problem parameters.

Since it is usually not possible to create an exam schedule which is perfect for everyone, to create the one that is, at least fair, to everyone is very important, especially in today's larger-size institutions with their intensive educational programs. Such an exam schedule, a balanced-exam schedule, should be preferable, we believe, since it not only allows people to balance their workloads, but also prevents the conflicts and disagreements among them. This study aims at introducing a novel approach for creating such a balanced-exam schedule. Future study directions include some extensions to our approach such as the consideration of proctor assignments and providing a graphical user interface for practical use.

Acknowledgments We would like to thank to the staff of the Faculty of the Engineering and the Department of Industrial Engineering of Uludag University on behalf of Erdal Emel, the department head, for the support provided.

References

1. Johnes, J.: Operations research in education. Eur. J. Oper. Res. **243**(3), 683–696 (2015)
2. Al-Yakoob, S.M., Sherali, H.D.: Mathematical programming models and algorithms for a class-faculty assignment problem. Eur. J. Oper. Res. **173**, 488–507 (2006)
3. Daskalaki, S., Birbas, T., Housos, E.: An integer programming formulation for a case study in university timetabling. Eur. J. Oper. Res. **153**, 117–135 (2004)
4. Daskalaki, S., Birbas, T.: Efficient solutions for a university timetabling problem through integer programming. Eur. J. Oper. Res. **160**, 106–120 (2005)
5. Philips, A.E., Waterer, H., Ehrgott, M., Ryan, D.M.: Integer programming methods for large-scale practical classroom assignment problems. Comput. Oper. Res. **53**, 42–53 (2015)
6. Al-Yakoob, S.M., Sherali, H.D.: Mathematical models and algorithms for a high school timetabling problem. Comput. Oper. Res. **61**, 56 (2015)
7. Alzaqebah, M., Abdullah, S.: Hybrid bee colony optimization for examination timetabling problems. Comput. Oper. Res. **54**, 142–154 (2015)
8. Burke, E.K., Newall, J.P.: Solving examination timetabling problems through adaption of heuristic orderings. Ann. Oper. Res. **129**, 107–134 (2004)
9. Carter, M.W., Laporte, G., Lee, S.Y.: Examination timetabling: algorithmic strategies and applications. J. Oper. Res. Soc. **47**(3), 373–383 (1996)
10. Babaei, H., Karimpour, J., Hadidi, A.: A survey of approaches for university course timetabling problem. Comput. Ind. Eng. (2014). doi:10.1016/j.cie.2014.11.010
11. Burke, E.K., Jackson, K., Kingston, J.H., Weare, R.F.: Automated university timetabling: the state of the art. Comput. J. **40**(9), 565–571 (1997)
12. Burke, E.K., Petrovic, S.: Recent research directions in automated timetabling. Eur. J. Oper. Res. **140**, 266–280 (2002)
13. Schaerf, A.: A survey of automated timetabling. Artif. Intell. Rev. **13**, 87–127 (1999)
14. Schmidt, G., Strohlein, T.: Timetable construction an annotated bibliography. Comput. J. **23**(4), 307–316 (1980)
15. Burke, E.K., Elliman, D.G., Ford, P.H., Weare, R.F.: Examination Timetabling in British Universities: A Survey. Practice and Theory of Automated Timetabling I, pp. 76–90. Springer, Berlin (1996)
16. Carter, M.W.: OR practice—a survey of practical applications of examination timetabling algorithms. Oper. Res. **34**(2), 193–202 (1986)
17. Carter, M.W., Laporte, G.: Recent Developments in Practical Examination Timetabling. Practice and Theory of Automated Timetabling I, pp. 3–21. Springer, Berlin (1996)
18. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T.G., Lee, S.Y.: A survey of search methodologies and automated system development for examination timetabling. J. Sched. **12**, 55–89 (2009)
19. Carter, M.W., Laporte, G., Chinneck, J.W.: A general examination scheduling system. Interfaces **24**, 109–120 (1994)
20. Al-Yakoob, S.M., Sherali, H.D., Al-Jazzaf, M.: A mixed-integer mathematical programming modeling approach to exam timetabling. Comput. Manag. Sci. **7**, 19–46 (2010)
21. Burke, E.K., Bykov, Y., Petrovic, S.: A Multicriteria Approach to Examination Timetabling. Practice and Theory of Automated Timetabling III, pp. 104–117. Springer, Berlin (2001)
22. Brailsford, S.C., Potts, C.N., Smith, B.M.: Constraint satisfaction problems: algorithms and applications. Eur. J. Oper. Res. **119**, 557–581 (1999)
23. MirHassani, S.A.: Improving paper spread in examination timetables using integer programming. Appl. Math. Comput. **179**(2), 702–706 (2006)
24. Anderson, J.M., Bernhard, R.H.: A university examination-scheduling model to minimize multiple-examination days for students. Decis. Sci. **12**(2), 231–239 (1981)
25. Ismayilova, N.A., Sagir, M., Gasimov, R.N.: A multiobjective faculty-course-time slot assignment problem with preferences. Math. Comput. Model. **46**, 1017–1029 (2007)
26. Ozdemir, M.S., Gasimov, R.N.: The analytic hierarchy process and multiobjective 0–1 faculty course assignment. Eur. J. Oper. Res. **157**, 398–408 (2004)
27. Zadeh, L.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
28. Nguyen, H.T.: On foundations of fuzzy theory for soft computing. Int. J. Fuzzy Syst. **8**(1), 39–45 (2006)
29. Azadegan, A., Porobic, L., Ghazinoory, S., Samouei, P., Kheirkhah, A.S.: Fuzzy logic in manufacturing: a review of literature and a specialized application. Int. J. Prod. Econ. **132**(2), 258–270 (2011)
30. Kar, S., Das, S., Ghosh, P.K.: Applications of neuro fuzzy systems: a brief review and future outline. Appl. Soft Comput. **15**, 243–259 (2014)
31. Kobbacy, K.A., Vadera, S., Rasmi, M.: AI and OR in management of operations: history and trends. J. Oper. Res. Soc. **58**, 10–28 (2007)
32. Melin, P., Castillo, O.: A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition. Appl. Soft Comput. **21**, 568–577 (2014)
33. Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. Manag. Sci. **17**(4), B-141–B-164 (1970)
34. Carlsson, C., Fuller, R.: Fuzzy multiple criteria decision making: recent developments. Fuzzy Sets Syst. **78**(2), 139–153 (1996)

35. Tao, Z., Chen, H., Zhou, L., Liu, J.: A generalized multiple attributes group decision making approach based on intuitionistic fuzzy sets. Int. J. Fuzzy Syst. **16**(2), 184–195 (2014)
36. Yu, D.: Group decision making under multiplicative hesitant fuzzy environment. Int. J. Fuzzy Syst. **16**(2), 233–241 (2014)
37. Tsai, C.M., Yeh, Z.M.: Classifying student's attentiveness via spatial-temporal fuzzy logic classification. Int. J. Fuzzy Syst. **16**(4), 541–553 (2014)
38. Asmuni, H., Burke, E.K., Garibaldi, J.M., McCollum, B.: An investigation of fuzzy multiple heuristics orderings in the construction of university exam timetables. Comput. Oper. Res. **36**(4), 981–1001 (2009)