Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 1584 – 1598

doi:10.3906/elk-1909-14

A spreadsheet-based decision support system for examination timetabling

Mehmet Güray GÜLER*, **Ebru GEÇİCİ**
Department of Industrial Engineering, Faculty of Mechanical Engineering, Yıldız Technical University, İstanbul, Turkey

Received: 02.09.2019 • **Accepted/Published Online:** 24.11.2019 • **Final Version:** 08.05.2020

Abstract: Examination timetabling is an inevitable problem of educational institutions. Each institution has its own particular limitations; however, the main structure is the same: assigning exams to time slots and classrooms. Several institutions solve the problem manually, but it becomes more difficult every year with increasing numbers of students and limited resources. There are many studies in the literature addressing the examination timetabling problem (ETP) and providing high quality solutions within reasonable amounts of time. Nevertheless, almost none of them can be used in practice since they are not converted into a decision support system (DSS). Commercial DSSs, on the other hand, are generally transactionally based and do not have optimization capabilities, i.e. they prevent conflicts via functional user interfaces. In this study, we propose a mixed integer programming (MIP) model that addresses the ETP of the Industrial Engineering Department of Yıldız Technical University. The model, which is capable of solving a wide range of similar ETP instances, is embedded into a DSS in the form of a spreadsheet. Given the enrollment lists of the courses, it generates schedules with minimum conflicts and consecutive exams while addressing requests of the lecturers and students. It does not require any technical knowledge and can be used by an average spreadsheet user. Moreover, it is flexible in terms of use for scheduling problems of other educational institutions. Currently, the DSS is in use by the department and real-life instances can be solved within a few seconds, saving significant amount of man-hours.

Key words: Examination timetabling, mixed integer programming, spreadsheet-based decision support system

1. Introduction

Examination timetabling is the art of allocating exams into appropriate time slots and classrooms while satisfying some constraints and personal preferences. Its structure is dependent on institutions. Many institutions do not have automatic scheduling software to create examination timetables and hence they are prepared manually, even in universities that have the ability to develop software themselves. The exam timetabling problem (ETP) is widely studied in undergraduate projects or graduate studies, which mainly focus on modeling the problem and/or proposing a solution method to a proposed model. However, there are limited number of studies that develop a decision support system (DSS). Therefore, most of the results are not applicable in terms of end users and become obsolete after the graduation of the student who prepared the research.

The Industrial Engineering Department (IE) of Yıldız Technical University (YTU) is one of the oldest industrial engineering programs in Turkey. The same problem applies to YTU-IE. Many undergraduate and master theses address the ETP of the department but timetables are still prepared manually. However, the student quota of the department has increased over the years while the number of classrooms remained constant

*Correspondence: mgguler@yildiz.edu.tr

and the number of teaching assistants (TAs) decreased gradually. Moreover, YTU has revised its curriculum: times of courses are changed and new courses are added. Currently it is very hard, if not impossible, to find a feasible timetable manually in a reasonable amount of time.

In this study, we present a spreadsheet-based DSS for preparing examination timetables and illustrate its use with the ETP of YTU-IE. First we propose a mixed integer programming (MIP) model that addresses constraints of the department, requests of the instructors, and convenience of the students. The model is prepared with an extended scope in such a way that it can be used for ETP instances of other universities. Then the MIP model is embedded into MS Excel using an add-in called Solver-Studio[1] to create the DSS. Using the enrollment lists, the DSS generates timetables in a reasonable amount of time and can be used by any average MS Excel user.

The rest of the paper is organized as follows. Section 2 provides a literature review for examination timetabling. Section 3 gives the problem statement. In Section 4, we give MIP model for the problem. We provide the details of the DSS and illustrate its implementation in Section 5. We conclude in Section 6.

2. Literature review

Educational timetabling for high school and university provides organization of classrooms and time in the best way for both students and instructors. It is generally divided into two groups: course timetabling and examination timetabling [1]. Course timetabling organizes courses in a term and creates programs that are followed weekly by avoiding conflicts in schedules. The general practice is to prepare a course timetable and not to change it unless needed. The need generally arises from non-routine activities like curriculum changes. In contrast to course timetabling, ETP is generally revised every term for several reasons. First, a crowded lecture may need more than one class for the examination, which may inhibit the exams of other courses. Therefore, one cannot use course timetables directly for examination timetables, especially for institutions with limited resources. Second, weekdays of holidays may change every year. For example, a holiday like January 1st that occurs on Monday can occur on Saturday in another year. Therefore, it takes time and effort to create or revise examination timetables. Third, the lecturer's preferences may change due to administrative tasks. ETP refers to two objects: the TA's assignment and exam scheduling (or examination timetabling). TA assignment is a variant of workforce planning and it assigns TAs to exams. Exam timetabling is the art of assigning exams to classrooms. In this study we focus on examination timetabling.

ETP is an NP hard problem [2] and various solution methods have been developed in the literature. There are different ways of classification for solution methods. For example, Qu et al. [3] classified solution approaches as graph-based techniques, constraint-based techniques, local-search techniques, population-based algorithms, multicriteria techniques, hyperheuristics, and decomposition. Gashgari et al. [1], on the other hand, categorized solution approaches as MIP, genetic algorithms, simulated annealing algorithms, quadratic assignments, and hybrid and tabu searches. It is observed that the number of articles that use metaheuristic approaches is greater than the studies dealing with exact solution methods [3]. Similarly, Babaei et al. [4] reported that most of the papers that deal with course timetabling use metaheuristic solution methods. Metaheuristic approaches constitute the majority of the studies due to their scalability, i.e., good solutions can be found in a reasonable amount of time. Although mathematical models like MIP models are able to find optimal solutions, they are used in few studies since their scalability is not comparable with metaheuristics. However, with the increasing

[1]Solver Studio(2019). About Solver Studio [online]. Website https://solverstudio.org/ [accessed 30 July 2019].

power of CPUs and the quality of the solvers, real-life problem instances can be solved in a reasonable amount of time using MIP models and commercial solvers. In this study, we develop a mathematical model (MIP), embed the model into a DSS, and use a commercial solver to solve the model. Therefore, we focus on the studies that model ETP with mathematical models and/or develop DSSs for ETPs or similar problems like the course timetabling problem (CTP).

The International Timetabling Competition provides a very general instance for the ETP. McCollum et al. [5] developed an integer programming (IP) model that addresses the ETP of the competition. This model is improved by preprocessing stages in the study of Arbaoui et al. [6]. The preprocessing stages reveal general conflict constraints. With the improved model, the numbers of hard and soft constraints were reduced. Gogos et al. [7] used a greedy randomized adaptive search procedure that involves several optimization algorithms together with an IP model.

The above papers provide solution methods for the same ETP instance. There are also several studies that address institution-specific problems. Al-Yakoob et al. [8] proposed two MIP models to solve ETP and TA assignment problems of Kuwait University. They developed a heuristic approach to reduce dissatisfaction levels of the TAs during their assignment problems. Cavdur and Kose [9] proposed a fuzzy logic model which generates parameters of a MIP that solves the ETP of the Uludağ University IE department. TA assignment was fulfilled using a greedy heuristic. Komijan and Koupaei [10] developed a binary model to solve postgraduate students' exam schedules in the IE department at Islamic Azad University.

The ETP instances of these studies were solvable within reasonable time limits using commercial solvers. On the other hand, some problems required extra effort to be solved, i.e., the solvers or CPU powers were not sufficient to solve them. Hence, authors came up with several solution techniques. Dimopoulou and Miliotis [11] made an early attempt to solve a real-life instance. They provided an IP model for the ETP of Athens University of Economics and Business and solved it using a heuristic solution, which generates an initial feasible solution first and then improves the solution by relaxing some constraints. Qu and Burke [12] showed that problems can be decomposed into an easy set and a hard set of problems. They stated that the basic idea of decomposition is to "divide and conquer", as optimal solutions of smaller subproblems may be much easier to obtain by using relatively simple approaches or even exact methods. They showed that although the difficult set is small in size, it makes a major contribution towards the total cost of the constructed solution. Later Qu et al. [13] used IP models to solve the hard subproblems of the ETP and introduced new cutting planes to have better solutions. In addition to these studies, Tilahun [14] created a heuristic approach to solve ETPs. He used a discrete version of the prey-predator algorithm and set up a simulation for tests.

Scheduling term-end exams of the United States Military Academy in West Point turns out to have different characteristics than usual ETPs. Wang et al. [15] addressed the ETP of West Point. They stated that there are hundreds of exams to schedule over such a short time period that there is simply no feasible solution. Hence, they allowed multiple sessions of the same exams and the aim was to minimize the number of duplicate exams. They had a two-stage solution method. In the first stage a good initial solution was developed using a greedy approach. This solution was improved in the second stage using MIP models. Two-stage methods turn out to be one of the common solution techniques. Lach et al. [16] developed a system to create course and exam schedules of the Technical University of Berlin and Keskin et al. [17] studied the ETP of the Engineering Faculty of Pamukkale University. Both studies developed a two-stage method where the first stage assigns time slots to the courses and the second stage assigns classrooms.

Only a few of the studies cited above were successful to develop a DSS to solve real-life instances. Wang et al. [15] developed a DSS together with GAMS Development Corporation. The system of Lach et al. [16] was first implemented at the Technical University of Berlin and then at RWTH Aachen University in 2013, and finally at the Technical University of Munich in 2015. Dimopoulou and Miliotis [11] developed a computer application for CTP and ETP at Athens University of Economics and Business.

The number of DSS studies related to ETP is limited. Hence, we also consider DSS studies proposed for other educational scheduling problems. In fact, DSS studies regarding CTP constitute the majority of the DSS studies in education. Piechowiak and Kolski [18] developed constraint programming for CTP and they implemented it by using multiagent approaches to create a DSS. Miranda [19] created a computer system, “eClasSkeduler”, by using IP for CTP. The author applied this system at Universidad de Chile. Then Miranda et al. [20] formed a web-based system, “udpSkeduler”, which uses a MIP model, for the CTP of Universidad Diego Portales. Al-Qaheri et al. [21] constructed a computer system for Kuwait University. Their DSS, which they created using an IP, consists of three stages: the faculty-course assignment stage, the course-time slots assignment stage, and time slot-room assignment stage.

We noticed two very recent studies that provided a DSS for educational purposes. Bailey and Michaels [22] provided a spreadsheet-based DSS for assigning students to teachers in primary schools and junior high schools in the United States. The problem is formulated with a MIP model and they solve the model in an MS Excel sheet using an open-source solver called Open Solver.[2] Siddiqui et al. [23], on the other hand, studied a multilevel problem which consists of course offerings, instructor assignments, and preparing course timetables. They provided a web-based DSS and implemented it at a Middle Eastern university. Table 1 shows ETP and CTP studies using mathematical programming methods and/or proposing DSS.

Emerging technologies (the Django framework for Python,[3] for example) facilitates development of web sites and web services. Similarly, many add-ins developed for MS Excel (like Open Solver or Solver Studio) have improved the functionality of MS Excel. Therefore embedding mathematical models and optimization into user-friendly DSSs is easier than before. This effect can be observed in the literature, as well [22, 23]. In this study, we present a DSS to solve the ETP of YTU-IE. The DSS is in the form of an MS Excel spreadsheet and can be easily used by an average MS Excel user. We developed a MIP model to address the ETP of the department and used Python to code this model in an add-in (of MS Excel) called Solver Studio. This MIP lies in the core of the DSS and it is able to solve the current instances within a few seconds. We circumvent the problem of defining constraints to prevent conflicts or consecutive exams by using enrollment lists. It is currently in use in the department for the last two semesters, saving significant (highly qualified) man-hour work with fewer complaints. The DSS and the MIP model are flexible in the sense that they can be easily adapted for solving ETPs of other universities.

3. Problem statement

In this section, we present the current structure of the department and define the problem. YTU-IE offers two programs at the undergraduate level: a Turkish (or 30% English) program and English (or 100% English) program. The students enrolled in the Turkish program have to complete 30% of their credit load with English courses. Hence, the students in Turkish programs can attend the courses offered in English as well. The courses are offered in two semesters: fall and spring. In 2018, there was an update to the curriculum of the department.

[2] OpenSolver(2019). About Open Solver [online]. Website https://opensolver.org/ [accessed 30 July 2019].
[3] Python(2019). Python [online]. Website https://www.python.org/ [accessed 30 July 2019].

Table 1. Classification of ETP and CTP with DSS.

Author(s)	Year	Problem	Solution method	DSS Status	DSS Type
Dimopoulou and Milotis [11]	2001	CTP + ETP	IP + heuristic	✓	A PC-based computer system
Piechowiak and Kolski [18]	2004	CTP	Constraint programming	✓	A computer system based on multiple agents
Qu and Burke [12]	2007	ETP	Decomposition		-
Qu et al.[13]	2009	ETP	IP + decomposition		-
Wang et al. [15]	2010	ETP	MIP + heuristic + decomposition	✓	A computer system
Al-Yakoob et al. [8]	2010	ETP	MIP + heuristic (for TA assignment)		-
Miranda [19]	2010	CTP	IP	✓	A computer system
Al-Qaheri et al. [21]	2011	CTP	IP	✓	A computer system
Miranda et al. [20]	2012	CTP	IP	✓	Web-based
Gogos et al. [7]	2012	ETP	IP + Heuristic		-
Komijan and Koupaie [10]	2012	ETP	Binary model		-
McCollum et al. [5]	2012	ETP	IP		-
Arbaoui et al. [6]	2015	ETP	MIP + decomposition		-
Cavdur and Kose [9]	2016	ETP	MIP + heuristic		-
Lanch et al. [16]	2016	ETP	IP + decomposition		-
Keskin et al. [17]	2018	ETP	Decomposition + MIP		-
Siddiqui et al. [23]	2018	CTP	MIP	✓	Web-based
Bailey and Michaels [22]	2019	Student to teacher assignment	IP	✓	Spreadsheet
Tiluhan [14]	2019	ETP	Heuristic		-

As a result, some new courses were added to the curriculum and semesters of some courses were switched. For example, operation research 1 (OR1) was offered as a second-year class in the old curriculum and is offered as a third-year course in the new one. Statistics was offered in the fall in the old curriculum and it is offered in spring in the new one. Eighty students enroll in the Turkish program and 40 students enroll in the English program every year. The total number of students adds up to 600 with students coming from double majors and exchange programs like Erasmus or Farabi. Despite this large number of undergraduate students, YTU-IE has only 5 classrooms to use for lectures and exams. Besides, the number of TAs decreased from 25 to 6 in the last three years. In fact, this is a general issue at all universities since the Higher Educational Institute of Turkey (YÖK) follows a strategy to reduce the number of TAs at the national level. In order to generate a feasible examination timetable several rules should be satisfied, which are listed below:

- Each exam must be assigned to a specific day and slot.
- There must be only one exam in a classroom for a slot.
- There are one midterm exam and one final exam in a term. We will use *midterm* and *final* for the midterm exam and final exam, respectively. There is a one-week break in each semester for midterms (called the *midterm week*). Finals are held at the end of the term in the so-called *final week*. Although the midterm week lasts one week, the final week can last more than a week. Therefore, it is not always possible to use the timetable of the midterm week for the final exams. Please note that the department can use only a few days since the remaining days are allocated to service courses like mathematics, physics, etc.
- Days are divided into six equal time slots that lasts ninety minutes.
- The undergraduate program can use five classrooms for examinations. Moreover, graduate courses continue in the midterm week as well and hence the number of available classrooms is generally smaller than five.
- Instructors may have more than one course and they may ask to have their examinations at the same time. Instructors also may choose specific days or slots for exams.
- If there are *significant* numbers of students enrolled in any two courses, then their exams must be in different time slots and their exams must be one slot apart, i.e., there should be 90 minutes between two consecutive exams.
- On the contrary, some courses' exams must be held simultaneously. For example, both Turkish and English OR1 courses have exactly the same content and their exams should be planned in the same time slot.
- Service course examinations are scheduled by other departments and their exams are held on another campus. Hence, no exams are planned on the examination days of service courses.
- Students may not have more than three exams in a day.
- Exams last for a single slot.

Currently, it is almost impossible to create examination schedules manually that addresses the rules above, due to increasing numbers of students, the changes in the curriculum, and the decrease in the number of TAs.

4. Proposed MIP model

In this section we propose a MIP model that addresses the problem given in the previous section. The parameters and variables of the MIP model are given in Table 2 and Table 3, respectively.

Table 2. The parameters.

Set	Definition
I	Set of courses
J	Set of classrooms
D	Set of days
T	Set of slots
R	Set of groups
i,i',i",i"'	Course index
j,j'	Classroom index
d	Day index
t,t'	Slot index
h_j	Capacity of classroom j
k_i	Number of students enrolled in course i
$I_2(n)$	Course tuples of size two that have at least n students in common
$I_3(n)$	Course tuples of size three that have at least n students in common
$I_4(n)$	Course tuples of size four that have at least n students in common
I_s	Set of tuples of size two courses that must have a simultaneous exam
I_{ns}	Courses that must not have a simultaneous exam
I_p	Courses of instructor p
I_r	Courses of student group r
θ_{rdt}	1 if classroom j is available on day d, slot t
α_{idt}	1 if course i should have an exam on day d at slot t
β_{idt}	1 if course i should *not* have an exam on day d at slot t
J_i^1	Set of eligible classrooms for examination of course i
J_{dt}^2	Set of eligible classrooms on day d, slot t
A_j	Required number of TAs for an examination in classroom j
AC_{dt}	Number of available TAs on day d at slot t
e_{idt}^1	Penalty value for variable Z_{idt}
e_{ijdt}^2	Penalty value for variable X_{ijdt}

Table 3. The variables.

Variable	Definition
X_{ijdt}	1 if exam i is in classroom j on day d at slot t, and 0 otherwise
Z_{idt}	1 if exam i is on day d at slot t, and 0 otherwise

We first propose our MIP model below and then explain the constraints and the objective function.

$$minz = \sum_{i \in I} \sum_{d \in D} \sum_{t \in T} Z_{idt} * d * \frac{k_i}{e_{idt}^1} + \sum_{i \in I} \sum_{j \in J} \sum_{d \in D} \sum_{t \in T} X_{ijdt} * e_{ijdt}^2,$$

subject to

$$\sum_{d\in D}\sum_{t\in T} Z_{idt} = 1, \quad i \in I, \tag{1}$$

$$\sum_{i\in I} X_{ijdt} \leq 1, \quad j \in J, d \in D, t \in T, \tag{2}$$

$$\sum_{j\in J} X_{ijdt} \leq 5 * Z_{idt}, \quad i \in I, d \in D, t \in T, \tag{3}$$

$$Z_{idt} \leq \sum_{j\in J} X_{ijdt}, \quad i \in I, d \in D, t \in T, \tag{4}$$

$$\sum_{d\in D}\sum_{t\in T}\sum_{j\in J}(h_j * X_{ijdt}) \geq k_i, \quad i \in I, \tag{5}$$

$$\sum_{d\in D}\sum_{t\in T} X_{ijdt} = 0, \quad i \in I, j \notin J_i^1, \tag{6}$$

$$Z_{idt} + Z_{i'dt} \leq 1, \quad (i,i') \in I_2(s_2), d \in D, t \in T, \tag{7}$$

$$Z_{idt} + Z_{i'dt+1} \leq 1, \quad (i,i') \in I_2(c_2), d \in D, t \in T, \tag{8}$$

$$Z_{idt} + Z_{i'dt+1} + Z_{i''dt+2} \leq 2, \quad (i,i',i'') \in I_3(c_3), d \in D, t \in T, \tag{9}$$

$$Z_{idt} + Z_{i'dt} + Z_{i''dt} \leq 1, \quad (i,i',i'') \in I_3(s_3), d \in D, t \in T, \tag{10}$$

$$\sum_{t\in T}(Z_{idt} + Z_{i'dt} + Z_{i''dt} + Z_{i'''dt}) \leq 3, \quad (i,i',i'',i''') \in I_4(d_4), d \in D, \tag{11}$$

$$Z_{idt} - Z_{i'dt} = 0, \quad (i,i') \in I_s, d \in D, t \in T, \tag{12}$$

$$Z_{idt} - Z_{i'dt} \leq 1, \quad (i,i') \in I_{ns}, d \in D, t \in T, \tag{13}$$

$$\sum_{i\in I_p} Z_{idt} \leq 1, \quad d \in D, t \in T, \tag{14}$$

$$\sum_{i\in I_r} Z_{idt} = 0, \quad d \in D, t \in T, r \in R \; s.t. \; \theta_{rdt} = 1, \tag{15}$$

$$X_{ijdt} = 0, \quad i \in I, j \in J, d = 3, t = 3, \tag{16}$$

$$Z_{idt} = 0, \quad i \in I, d \in D, t \in T \ s.t. \ \alpha_{idt} = 1, \tag{17}$$

$$Z_{idt} = 1, \quad i \in I, d \in D, t \in T \ s.t. \ \beta_{idt} = 1, \tag{18}$$

$$X_{ijdt} = 0, \quad i \in I, j \notin J^2_{dt}, d \in D, t \in T, \tag{19}$$

$$\sum_{i \in I} \sum_{j \in J} X_{ijdt} * A_j \leq AC_{dt}, \quad d \in D, t \in T, \tag{20}$$

$$X_{ijdt}, \ Z_{idt} \in \{0, 1\}, \quad i \in I, j \in J, d \in D, t \in T. \tag{21}$$

Constraint (1) ensures that each exam is scheduled at a time slot. Constraint (2) ensures that there is only one exam in a classroom on a given day and slot. Two different variables, X_{ijdt} and Z_{idt}, are used in the MIP model. Constraint (3) and constraint (4) provide the connection between these variables. While constraint (3) shows that a maximum of 5 classrooms are available, constraint (4) indicates that an exam given on a day and slot must be assigned to at least one classroom. In order to ensure that the exams are assigned to a sufficient number of classrooms, the capacity constraint (5) is used. Constraint (6) allows assignment of exams to a feasible classroom. Constraint (7) ensures that only one exam can be assigned to a student on a specific day and slot, i.e., it prevents conflicts. Constraint (8) is used to prevent students from taking two consecutive exams. However, there are some students who are studying in the department for more than 5 years and they attend many courses of different grades simultaneously.[4] Those students make it impossible to generate a schedule with constraint (7) and/or constraint (8) if simultaneous exams for courses that have at least 1 common student are avoided. Therefore, we define the set $I_2(s_2)$ in a parametric way so that the set includes the tuples of size two that have at least s_2 students so that $s_2 \geq 1$. Therefore, if $s_2 = 2$, it means that courses that have a single student in common can have simultaneous exams. This is allowed since, as stated before, there are no feasible schedules with $s_2 = 1$. Students cannot take three consecutive exams with constraint (9). Constraint (10) avoids three simultaneous exams for courses that have at least s_3 ($s_3 \geq 1$) common students. This constraint is needed when $s_2 \geq 2$ since no students have two simultaneous exams (and hence three simultaneous exams) if $s_2 = 1$. For our all instances, we were able to solve the problems with $c_3 = 1$, (i.e., nobody has three consecutive exams); hence, we did not add a constraint that avoids four consecutive exams. Constraint (11) is used to avoid students taking more than three exams in a day. Please note that one can use each year's courses to avoid consecutive and simultaneous exams, i.e., to implement constraints (7)–(11). For example, constraint (8) can be written so that a second-year student cannot have consecutive exams. However, with the new curriculum, several courses can be labeled with more than one year. For example, the statistics course was a third-year course but it is a second-year course in the current curriculum and hence both second-year and third-year students are enrolled in the statistics course. Moreover, some elective courses can be selected

[4] In Turkey, students are expelled from university if they cannot graduate after 7 years according to a law established in 2014. The students that are registered to universities before 2014 cannot be expelled.

in the second, third, or fourth year. Labeling these courses with three different years increases the probability of infeasibility. In fact, we have tried several problem instances with this setting but no feasible solution was found. We circumvent this problem by finding the number of common students for all combinations of two, three, and four courses using the enrollment lists from the registration database of the university. Then we create the lists $I_2(s_2)$, $I_2(c_2)$, $I_3(s_3)$, $I_3(c_3)$, and $I_4(d_4)$ and use them in the constraints. Recall that the lists show the number of common students of two, three, or four courses.

Some examinations must be held simultaneously. This is addressed by constraint (12). In contrast to constraint (12), some examinations should not be held simultaneously. This is ensured with constraint (13) and (14). Constraint (15) is used to avoid conflicts with service courses for which examination periods are predetermined. Constraints (16)–(18) ensures that examinations are assigned to required slots. The requirement can be dictated by the department or by the instructor. Constraint (19) avoids assigning of an exam to the classrooms that are not available. Constraint (20) handles the TA constraint.

The objective function has three pillars. First, the academic staff wants the exams to finish as early as possible to spare time for grading the exams. Therefore, the model schedules the classes (especially the crowded ones) as early as possible. Second, there is a significant decrease in the success of the students in late exams. Third, it minimizes the number of assigned classrooms, which also corresponds to minimizing the number of TAs in charge. The first part in the objective function deals with early finishes and last slots through the coefficients e^1_{idt}. The second part handles the number of classrooms.

5. DSS implementation

In this section, we first describe our DSS and then we illustrate the implementation with the timetables of spring and fall semesters of 2018–2019 for YTU-IE.

The aim of this study is to create a DSS that enables to prepare timetables addressing the rules of the department, the requests of instructors, and the needs of the students. We used MS Excel and its add-in Solver Studio to develop the DSS. Solver Studio contains an interpreter that allows coding using different programming languages. We used Python for coding and Gurobi[5] for solving the proposed MIP model.

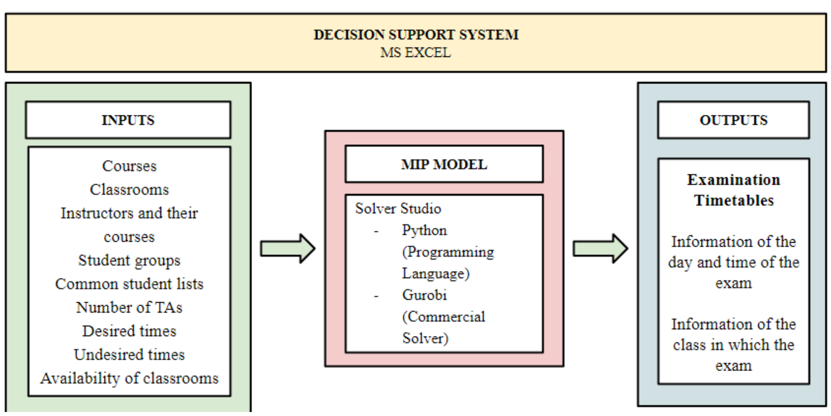

Figure 1. Structure of DSS.

[5]Gurobi(2019). Gurobi [online]. Website https://www.gurobi.com/ [accessed 30 July 2019].

The structure of the DSS, which consists of three parts, is given in Figure 1. The first part is the system input. Types of input data are given in 4. These tables are created by using the data items editor of Solver Studio in Figure 2. They are the parameters that are required for running the MIP model. They consist of courses, classrooms, student groups, desired periods, and undesired periods. The second part consists of the MIP model that solves the problem. The output is the last part. All inputs and outputs are written into the cells of MS Excel in forms of tables.

Table 4. The information used in the input area.

1.Courses and course capacities	7.Instructors' courses
2.Number of days and slots	8.Classrooms and classroom capacities
3.Desired days/slots	9.Student groups
4.List of courses (of size two, three, and four)	10.Undesired days/slots
5.Availability of the classrooms	11.Number of available TAs
6.A list of simultaneous courses	

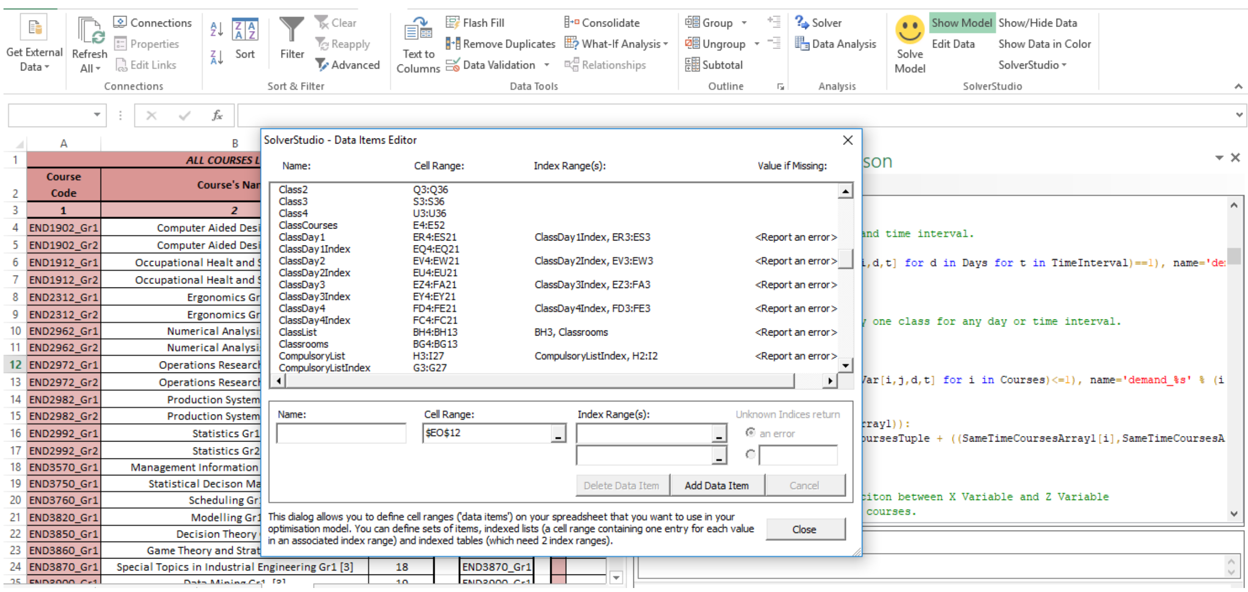

Figure 2. Input area – data items editor of Solver Studio.

In the spring semester of 2018–2019, $I_2(s_2 = 1)$ has 758, $I_3(s_3 = 1)$ has 3228, and $I_4(d_4 = 1)$ has 5033 elements. Numbers of common students for these lists are given in Figure 3. Each slice in the pie charts gives the number of tuples created according to the number of common students. In Figure 3a, for example, there are 107 pairs with 1 common student and 338 pairs that have 6 or more common students. It can be observed that using the set with $c_2 = 2$ rather than $c_2 = 1$ removes 107 constraints from the problem. Similarly, using $d_4 = 2$ rather than $d_4 = 1$ removes 3659 constraints. Like Figure 3a, Figure 3b and Figure 3c give the number of tuples of the common students in three and four comparison lists, respectively.

The instances were run on an Intel Core i5-3470 CPU computer with 4 GB RAM. We prepared the midterms and the finals for fall and spring semesters of 2018–2019. We set the optimality gap to 10%, which is found to work well during experiments. Computational results of spring term examination timetables are given

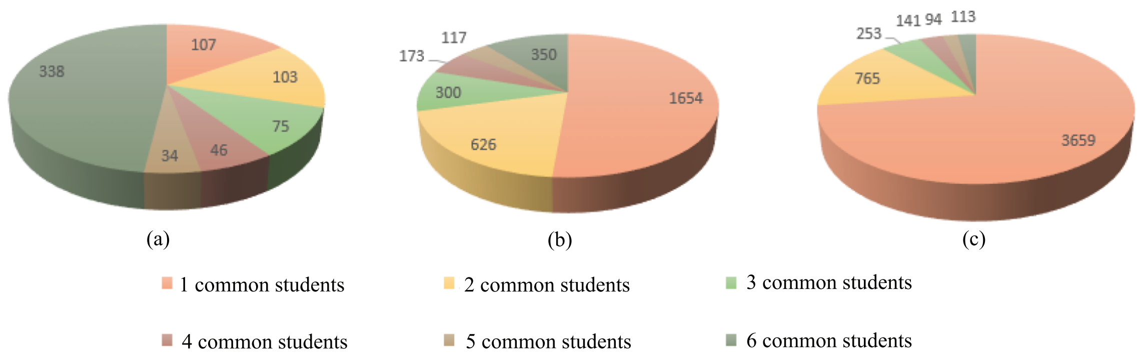

Figure 3. Number of common students for tuples of sizes of (a) two courses, (b) three courses, (c) four courses.

in Table 5. Since the final program has more issues to be addressed, the number of constraints and variables is more in the final program. It took less than 20 seconds to create the exam schedules for both semesters using the DSS. We announced the resulting timetables on the departmental website and received feedback from both students and academics to revise our model and the DSS.

Table 5. Computational results of the MIP for spring term in 2018–2019.

Exam	Rows	Columns	Integer	Nonzero entries	Iterations	Solution time (s)
Midterm	204795	8640	8640	671547	42982	8.08
Final	268814	15840	15840	975688	78562	17.28

One motivation in this study is the increasing number of complaints from students due to conflicts, i.e., having simultaneous exams. Note that we handled this issue with constraints (7) and (12) through the sets $I_2(.)$ and $I_3(.)$ Table 6 shows a comparison of conflicts in midterm exam schedules prepared manually and by DSS for spring midterms. *Exam* indicates the number of exams in one term and *conflicts* shows the number of students that have conflicts in their exams. At YTU, students must have a legal excuse (medical report, conflict, etc.) to take a make-up exam for the midterm. On the other hand, no excuse is required to take the re-sit exam (i.e., the make-up exam for the final). Therefore, the number of students with conflicts is recorded only for midterms by the department secretariat. It can be observed that the number of students with conflicts was reduced, although the number of examinations increased. In order to have a normalized measure, we divided the number of students with conflicts into the number of exams. This ratio shows the number of students that have a conflict per exam. The DSS reduced the ratio by 22%.

Table 6. The midterm conflict-related data of YTU-IE.

Year	Source	Exam	Conflicts	Conflict ratio
2017–2018	Manual	43	30	0.70
2018–2019	DSS	48	26	0.54

The sets $I_2(.)$, $I_3(.)$, and $I_4(.)$ are characterized by the parameters s_2, c_2, s_3, c_3, and d_4. These parameters affect the feasibility and quality of the solution through the sets. We illustrate their effect throughout the final timetable of 2018 Fall. Generally, the final week lasts two weeks (or 10 days). This period includes service course examinations (3.5 days) and presentations of graduation theses of the fourth-year students (2 days). Therefore, 4.5 days are left in two weeks for YTU-IE to allocate the finals of the department. In order to see the timetables for different lengths of periods, we tried some combinations of different days and slots. For each day-slot combination, we tuned the parameters s_2, c_2, s_3, c_3, and d_4 to find a feasible solution. We prepared the schedules for 3, 4, and 4.5 days with 5, 6, and 7 slots. These parameters and comparative results of these combinations are given in Table 7. Slot 6 starts at 16.30 and slot 7 starts at 18.00; hence, we tried to avoid allocating exams to these last two slots due to its negative effect on the overall success of the students. Moreover, the academic staff asks to finish the exams as early as possible to finish grading in a timely manner. We report the number of exams on consecutive days (1st day, 2nd day, etc.), the number of students taking exams in the first and second week, and the number of exams and students in the last slots, i.e., in slots 6 and 7 . We do not report solution time since all examination timetables are obtained in a minute. We do not give the number of classes assigned to exams since they are almost the same in every instance. It turns out that the numbers of exams are distributed evenly except the 3-day schedules, where the last day has significantly fewer exams. Since there are many students taking exams in slots 6 and 7, three-day exam schedules with 6 and 7 slots and a four-day schedule with 6 slots turn out to be dominated by other combinations. For the scenarios with 5 lost, the numbers of students with conflicting exams are quite high for the exam timetables with 3 days and 4 days. Therefore, these cases have been eliminated since the number of students whose exams overlap is high. Although the parameter values at 4.5 days of exams are not as high as 3 days and 4 days, the number of students in the first week is smaller than 6 and 7 slots. Thus, 4.5 days of exams with 5 slots dominates. For the remaining cases, the numbers of students in slots 6 and 7 are very close to each other. Moreover, students, TAs, and instructors do not want to have 7 slots unless it needed to maintain feasibility. The number of students taking exams in the first week turns out to be larger in the eighth combination (4.5 days with 6 slots). Note that this is possible since the values of the parameters change in each setting (Table 7). Using the results in Table 7 and following the confirmation of the department management, a 4.5-day schedule with 6 slots was announced as the final schedule. The final schedule is printed in the output tab of the spreadsheet and it can be directly saved as a PDF file to announce on the web.

Table 7. Comparative results for the final exams of spring term in 2018–2019.

Case no.	Indices		Parameters					Number of exams					No. of students		Last slot counts	
	Day	Slot	s_2 Sim. 2 exams	c_2 Con. 2 exams	s_3 Sim. 3 exams	c_3 Cons. 3 exams	d_4 M. than 3 exams	Day 1	Day 2	Day 3	Day 4	Day 5	Week 1	Week 2	Exam	Student
1	3	5	11	10	10	10	10	20	16	12	-	-	2888	-	-	-
2	3	6	6	5	5	5	5	18	18	12	-	-	2888	-	10	652
3	3	7	4	2	2	2	2	16	20	12	-	-	2888	-	11	615
4	4	5	4	3	3	3	3	13	15	9	11	-	2165	723	-	-
5	4	6	1	3	1	3	2	12	13	12	11	-	2168	720	9	635
6	4	7	1	2	1	2	2	11	14	11	12	-	2092	796	11	538
7	4,5	5	2	3	3	2	2	12	12	11	9	4	2081	807	-	-
8	**4,5**	**6**	**1**	**3**	**1**	**3**	**2**	**11**	**12**	**8**	**12**	**5**	**2163**	**725**	**8**	**546**
9	4,5	7	1	2	1	2	2	12	12	9	12	3	2024	864	7	515

6. Conclusion and future work

We proposed a DSS to solve the ETP of YTU-IE. We addressed the ETP with a MIP model and embedded the MIP in the core of the DSS, which is in the form of an MS Excel sheet. Given the enrollment lists of the courses, the DSS is able to generate examination timetables in a few seconds. The DSS does not require any coding or optimization expertise and can be used by an average MS Excel user by fulfilling the requests and requirements of the department, instructors, and students. The DSS is able to prepare examination timetables with a reduced exam intensity in a day. Moreover, a balanced program can be created by providing the best way to ensure that students do not have an exam at the same time, no two consecutive exams, and no more than three exams in a day. Since all the required information is defined in the DSS, a system-oriented structure has been created by preventing loss of information. Thus, it is always possible to create schedules with certain standards for which the same rules are followed.

We keep the scope of the MIP model and the DSS as general as possible to be able to solve similar ETPs of other departments and institutions. Anyone with basic Python knowledge can modify the DSS to change the MIP model easily to address any similar scheduling problems. Indeed, we made a small modification to the current version to address the ETP of the Vocational High School of İstanbul University, which has 7 departments and more than 2000 students. Moreover, the Mechatronics Engineering Department of YTU plans to use the DSS for their examination timetables in the forthcoming semester without any change since their ETP is almost the same as YTU-IE.

There are two future directions to be explored. First, the current version uses commercial software, Gurobi, which can solve the current instances in a reasonable amount of time. However, for larger instances, solution time can be beyond tolerable limits. Currently, the fall midterm schedule of the Vocational High School of İstanbul University is being prepared by a modified version of our DSS. It takes 30 more than 2 hours to find a schedule. Therefore, our aim is to use a heuristic method (for example, the two-stage method of [17]) or develop a new method to solve the problem fast. Second, we want to develop a website so that the DSS can be used on any computer without any required installation.

References

[1] Gashgari R, Alhashimi L, Aljawi L, Alamoudi A. A survey on exam scheduling techniques. In: 2018 1st International Conference on Computer Applications & Information Security (ICCAIS); Riyadh, Saudi Arabia; 2018. pp. 1-5.

[2] Arogundade OT, Akinwale AT, Aweda OM. A genetic algorithm approach for a real-world university examination timetabling problem. International Journal of Computer Applications 2010; 12 (5): 0975-8887.

[3] Qu R, Burke EK, McCollum B, Merlot LTG, Lee SY. A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling 2009; 12 (1): 55-89.

[4] Babaei H, Karimpour J, Hadidi A. A survey of approaches for university course timetabling problem. Computers & Industrial Engineering 2015; 86: 43-49.

[5] McCollum B, McMullan P, Parkes AJ, Burke EK, Qu R. A new model for automated examination timetabling. Annals of Operations Research 2012; 194 (1): 291-315.

[6] Arbaoui T, Boufflet JP, Moukrim A. Preprocessing and an improved MIP model for examination timetabling. Annals of Operations Research 2015; 229 (1): 19-40.

[7] Gogos C, Alefragis P, Housos E. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. Annals of Operations Research 2012; 194 (1): 203-221.

[8] Al-Yakoob SM, Sherali HD, Al-Jazzaf M. A mixed-integer mathematical modeling approach to exam timetabling. Computational Management Science 2010; 7 (1): 19-46.

[9] Cavdur F, Kose M. A fuzzy logic and binary-goal programming-based approach for solving the exam timetabling problem to create a balanced-exam schedule. International Journal of Fuzzy Systems 2016; 18 (1): 119-129.

[10] Komijan AR, Koupaei MN. A new binary model for university examination timetabling: a case study. Journal of Industrial Engineering International 2012; 8 (1): 28.

[11] Dimopoulou M, Panagiotis M. Implementation of a university course and examination timetabling system. European Journal of Operational Research 2001; 130 (1): 202-213.

[12] Qu R, Burke EK. Adaptive decomposition and construction for examination timetabling problems. In: Proceedings of the 3rd Multidisciplinary International Scheduling: Theory and Applications; Paris, France; 2007. pp. 418-425.

[13] Qu R, He F, Burke EK. Hybridizing integer programming models with an adaptive decomposition approach for exam timetabling problems. In: The 4th Multidisciplinary International Scheduling: Theory and Applications; Dublin, Ireland; 2009. pp. 435-446.

[14] Tilahun SL. Prey-predator algorithm for discrete problems: a case for examination timetabling problem . Turkish Journal of Electrical Engineering & Computer Sciences 2019; 27 (2): 950-960.

[15] Wang S, Bussieck M, Guignard M, Meeraus A, O'Brien F. Term-end exam scheduling at United States Military Academy/West Point. Journal of Scheduling 2010; 13 (4): 375-391.

[16] Lach G, Lach M, Steenken J, Zorn E. Examination timetabling with Moses: System demonstration. In: Proceedings of the 11th International (Confenference on Practice and Theory of Automated Timetabling (PATAT" 16); Udine, Italy; 2016. pp. 511-514.

[17] Keskin ME, Döyen A, Akyer H, Güler MG. Examination timetabling problem with scarce resources: a case study. European Journal of Industrial Engineering 2018; 12 (6): 855-874.

[18] Piechowiak S and Kolski C. Towards a generic object oriented decision support system for university timetabling: an interactive approach. International Journal of Information Technology & Decision Making 2004; 3 (1): 179-208.

[19] Miranda J. eClasSkeduler: a course scheduling system for the executive education unit at the Universidad de Chile. Interfaces 2010; 40 (3): 196-207.

[20] Miranda J, Rey PA, Robles JM. udpSkeduler: A web architecture based decision support system for course and classroom scheduling. Decision Support Systems 2012; 52 (2): 505-513.

[21] Al-Qaheri H, Hasan MK, Al-Husain R. A decision support system for a three-stage university course scheduler with an application to College of Business Administration, Kuwait University. International Journal of Data Analysis and Information Systems 2011; 3 (2): 95-110.

[22] Bailey MD, Michaels D. An optimization-based DSS for student-to-teacher assignment: Classroom heterogeneity and teacher performance measures. Decision Support Systems 2019; 119: 60-71.

[23] Siddiqui AW., Raza SA, Tariq ZM. A web-based group decision support system for academic term preparation. Decision Support Systems 2018; 114: 1-17.