A spreadsheet based decision support system for examination timetabling

Mehmet Güray GÜLER¹*, Ebru GEÇİÇİ²

¹,²Industrial Engineering Department, Faculty of Mechanical Engineering, Yıldız Technical University, İstanbul, Turkey

Abstract: Examination timetabling is an inevitable problem of educational institutions. Each institution has its own particular limitations, however, the main structure is same: assigning exams to time slots and classrooms. Several institutions solve the problem manually, but it becomes more difficult every year with increasing number of students and limited resources. There are many studies in the literature addressing Examination Timetabling Problem (ETP) and provide high quality solutions within reasonable amount of time. Nevertheless, almost none of them can be used in practice since they are not converted into a decision support system (DSS). Commercial DSS’s, on the other hand, are generally transactional 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 enrolment 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’s flexible in terms of using 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 an automatic scheduling software to create examination timetables and hence they are prepared manually, even in universities that has the ability to develop a software by itself. In fact, the exam timetabling problem (ETP) is widely studied in undergraduate projects or graduate studies, which mainly focus on either modeling the problem and/or proposing a solution method to a proposed model. However, there is only a few number of research that develops 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.

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 graduation thesis and master thesis address the ETP of the department but timetables are still prepared manually. However, student quota of department increased over the years, on the contrary number of classroom remained constant and the number of teaching assistants (TA) decreased gradually. Moreover, YTU has revised its curriculum: time of

*Correspondence: mgguler@yildiz.edu.tr
courses are changed and new courses are added. Currently it's 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 ETP of YTU-IE department. 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 enrolment 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 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 \cite{1}. Course timetabling organizes courses at 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 it is needed. The need generally arises from non-routines activities like curriculum changes.
In contrast to course timetabling, ETP is generally revised every term due to several reasons. First, a crowded
lecture may need more than one class for the examination which may inhibit making exams of other courses.
Therefore one cannot use course timetables directly for examination timetables, especially for institutions with
limited resources. Second, the weekday of holidays may change every year. For example a holiday like January
1\(^{st}\) 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: 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 \cite{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. \cite{3} classified solution
approaches as graph-based techniques, constraint-based techniques, local-search techniques, population based
algorithms, multi-criteria techniques, hyper-heuristics and decomposition. Gashgari et.al. \cite{1}, on the other
hand, categorized solution approaches as MIP, genetic algorithm, simulated annealing algorithm, quadratic
assignment, and hybrid and tabu search. It is observed that the number of articles that uses meta heuristic
approaches is more than the studies dealing with exact solution methods \cite{3}. Similarly, Babaei et.al.\cite{4} reports
that most of the papers that deal with course timetabling use meta-heuristic solution methods. Meta-heuristic
approaches constitute majority of the studies due to their scalability, i.e., good solution 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 a few number of studies since their scalability is not comparable with meta-heuristics. However,
with increasing power of CPUs and 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 (i) we develop a mathematical model

\(^{1}\)Solver Studio(2019).About Solver Studio [online].Website https://solverstudio.org/ [accessed 30 July 2019]
(an MIP), (ii) embed the model into a DSS, and (iii) use a commercial solver to solve the model. Therefore, we focus on the studies that model ETP with mathematical models and/or develop DSS for ETP or similar problems like course timetabling problem (CTP).

International Timetabling Competition provides a very general instance for 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 study of Arbaoui et.al. [6]. The preprocessing stages reveal general conflict constraints. With the improved model, the number of hard and soft constraint were reduced. Gogos et.al. [7] used 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 heuristic approach to reduce dissatisfaction levels of the TAs for during their assignment problem. Cavdur and Kose [9] proposed a fuzzy logic model which generates parameters of a MIP that solves ETP of 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 of the IE department at Islamic Azad University.

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 power were not sufficient to solve them. Hence authors came up with several solution techniques. Dimopoulos and Miliotis [11] is an early attempt to solve a real life instance. They provided an IP model for ETP of Athens University of Economics and Business and solve it using a heuristic solution which generates an initial feasible solution first and then improves the solution by relaxing some constraint. 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 sub-problems 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 sub problems 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 ETP. He used a discrete version of the the prey-predator algorithm and set up a simulation for test.

Scheduling term-end exams of United States Military Academy in West Point turns out to have different characteristics than usual ETP. Wang et.al. [15] addressed 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 Technical University of Berlin and Keskin et.al.[17] studied the ETP of Engineering Faculty in 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 Technical University of Berlin and then at RWTH Aachen University in 2013 and finally
at Technical University of Munich in 2015. Dimopoulou and Miliotis [11], developed a computer application based for CTP and ETP in the Athens University of Economics and Business.

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

We noticed two very recent studies that provides DSS for educational purposes. Bailey and Michaels [22] provides a spreadsheet based DSS for assigning students to teachers in primary schools and middle high schools in United States. The problem is formulated with a MIP model and solve the model in an MS Excel sheet using an open source solver called Open Solver². Siddiqui et. al. [23], on the other hand, studied a multi-level problem which consists of course offerings, instructor assignments and preparing course timetables. They provide a web based DSS and implement it at a Middle Eastern University. Table 1 shows ETP and CTP studies using the mathematical programming methods and/or proposing DSS.

Emerging technologies (Django framework for Python³, 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 DSS 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 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 enrolment lists. It is currently in use in the department for the last two semesters saving significant (highly qualified) man-hour work with less complaints. The DSS and the MIP model are flexible in the sense that they can be easily adapted for solving ETP 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: Turkish (or 30% English) program and English (or 100% English) program. The students enrolled to the Turkish program has 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 in the curriculum of the department. As a result, some new courses are added to the curriculum and semesters of some courses are 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, for example, was offered in the fall in the old curriculum and

---

²OpenSolver(2019). About Open Solver [online]. Website https://opensolver.org/ [accessed 30 July 2019]
³Python(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                          |
|-------------------|------|------------------|----------------------------------|------------------------------|
| Dimopoulou and    | 2001 | CTP + ETP        | IP + heuristic                   | ✓ A PC-based computer system |
| Miliotis [11]     |      |                  |                                  |                              |
| Piechowiak and    | 2004 | CTP              | Constraint Programming           | ✓ A computer system          |
| Kolski [18]       |      |                  |                                  | based on multi-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          |
|                   |      |                  | (for TA assignment)              |                              |
| Al-Yakoob et. al. | 2010 | ETP              | MIP + heuristic                  | -                            |
| [8]               |      |                  |                                  |                              |
| Miranda [19]      | 2010 | CTP              | IP                               | ✓ A computer system          |
| Al-Qaheri et. al. | 2011 | CTP              | IP                               | ✓ A computer system          |
| [21]              |      |                  |                                  |                              |
| Miranda et. al.   | 2012 | CTP              | IP                               | ✓ Web-based                  |
| [20]              |      |                  |                                  |                              |
| Gogos et.al. [7]  | 2012 | ETP              | IP + Heuristic                   | -                            |
| Komijan and       | 2012 | ETP              | Binary Model                     | -                            |
| Koupae [10]       |      |                  |                                  |                              |
| McCollum et. al.  | 2012 | ETP              | IP                               | -                            |
| [5]               |      |                  |                                  |                              |
| Arbaoui et.al. [6]| 2015 | ETP              | MIP + decomposition              | -                            |
|                   |      |                  |                                  |                              |
| Cavdur and Kose   | 2016 | ETP              | MIP + heuristic                  | -                            |
| [9]               |      |                  |                                  |                              |
| Lanch et. al. [16]| 2016 | ETP              | IP + decomposition               | -                            |
| Keskin et. al. [17]| 2018| ETP              | Decomposition + MIP              | -                            |
| Siddiqui et. al.  | 2018 | CTP              | MIP                              | ✓ Web-based                  |
| Bailey and        | 2019 | Student to teacher assignment | IP | ✓ Spreadsheet |
| Michaels [22]     |      |                  |                                  |                              |
| Tiluhan [14]      | 2019 | ETP              | Heuristic                        | -                            |

it is offered in spring in the new one. 80 students enroll to Turkish program and 40 students enroll to English program every year. Total number of students add up to 600 with students coming from double major and exchange programs like Erasmus or Farabi. Despite to this large number of undergraduate students, YTU-IE has only 5 classrooms to use for lectures and exams. Besides, number of TA decreased from 25 to 6 in the last three years. In fact this is a general issue in all universities since 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 midterm exam and final exam, respectively. There is a one week break in each semester for midterms (called the midterm week). Finals are made at the end of the term in the so called final week. Although midterm week lasts one week, final week can last more than a week. Therefore it’s not always possible to use a timetable of midterm week for the final exams. Please note that the department can use only a few days since 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 examination. Moreover, graduate courses continue in the midterm week as well, hence the number of available classrooms is generally smaller than five.

• Instructor may have more than one course and they may ask to have their examination at the same time. Instructors also may choose specific days or slots for exams.

• If there are significant number of students enrolled in any two courses, then (i) their exams must be on different time slots and (ii) 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 made simultaneously. For example both Turkish and English OR1 courses have exactly the same content and their exam should be planned at the same time slot.

• Service course examinations are scheduled by other departments and their exams are held in 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 a single slot.

Currently, it is almost impossible to create examination schedules manually that addresses the rules above, due to increasing number of students, the change 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.

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

\[
\min z = \sum_{i \in I} \sum_{d \in D} \sum_{t \in T} Z_{idt} \cdot d \cdot e_{idt} + \sum_{i \in I} \sum_{j \in J} \sum_{d \in D} \sum_{t \in T} X_{ijdt} \cdot e_{ijdt}^2
\]

Subject to

\[
\sum_{d \in D} \sum_{t \in T} Z_{idt} = 1, \quad i \in I
\]
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 has at least $n$ students in common |
| $I_3(n)$ | Course tuples of size three that has at least $n$ students in common |
| $I_4(n)$ | Course tuples of size four that has 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 $r$                   |
| $\theta_{rdt}$ | 1 if classroom $j$ is available on day $d$, slot $t$ |
| $\alpha_{idt}$ | 1 if course $i$ should have an exam on day $d$ at slot $t$ |
| $\beta_{idt}$ | 1 if course $i$ should not have an exam on day $d$ at slot $t$ |
| $J_1^i$ | Set of eligible classrooms for examination of course $i$ |
| $J_2^i$ | 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 $Z_{idt}$ |
| $e_{idt}^2$ | Penalty value for variable $X_{ijdt}$ |

Table 3. The variables

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

1. \[ \sum_{i \in I} X_{ijdt} \leq 1, \quad j \in J, d \in D, t \in T \] (2)

2. \[ \sum_{j \in J} X_{ijdt} \leq 5 \times Z_{idt}, \quad i \in I, d \in D, t \in T \] (3)

3. \[ Z_{idt} \leq \sum_{j \in J} X_{ijdt}, \quad i \in I, d \in D, t \in T \] (4)

4. \[ \sum_{d \in D} \sum_{t \in T} \sum_{j \in J} (h_j \times X_{ijdt}) \geq k_i, \quad i \in I \] (5)
\[ \sum_{d \in D} \sum_{t \in T} X_{ijdt} = 0, \ i \in I, j \notin J_i \] (6)

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

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

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

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

\[ \sum_{i \in I} (Z_{idt} + Z_{i'dt} + Z_{i''dt} + Z_{i''''dt}) \leq 3, \ (i, i', i'', i''') \in I_4(d_4), d \in D \] (11)

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

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

\[ \sum_{i \in I} Z_{idt} \leq 1, \ d \in D, t \in T \] (14)

\[ \sum_{i \in I} Z_{idt} = 0, \ d \in D, t \in T, r \in R \ s.t. \ \theta_{rdt} = 1 \] (15)

\[ X_{ijdt} = 0, \ i \in I, j \in J, d = 3, t = 3 \] (16)

\[ Z_{idt} = 0, \ i \in I, d \in D, t \in T \ s.t. \ \alpha_{idt} = 1 \] (17)

\[ Z_{idt} = 1, \ i \in I, d \in D, t \in T \ s.t. \ \beta_{idt} = 1 \] (18)

\[ X_{ijdt} = 0, \ i \in I, j \notin J_i^2, d \in D, t \in T \] (19)

\[ \sum_{i \in I} \sum_{j \in J} X_{ijdt} \ast A_j \leq A_C dt, \ d \in D, t \in T \] (20)

\[ X_{ijdt}, Z_{idt} \in \{0, 1\}, \ i \in I, j \in J, d \in D, t \in T \] (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 in a given day and slot. Two different variables, $X_{ijd}$ 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 maximum 5 classrooms is available, constraint (4) indicates that an exam which is given in 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 in a specific day and slot, i.e., it prevents conflicts. Constraint (8) is used to prevent students 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. 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 has 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 it is 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 didn’t add a constraint which avoids four consecutive exams. Constraint (11) is used to avoid students from 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 the constraints in (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, statistics course was a third year course but it is a second year course in the current curriculum, hence both second year and third year students are enrolled in the statistics course. Moreover some elective courses can be selected by 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 enrolment 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 made simultaneously. This is addressed by constraint (12). In contrast to constraint (12), some examinations should not be made simultaneously. This is ensured with constraint (13) and (14). Constraint (15) is used to avoid conflicts with service courses of which examination periods are predetermined. Constraint (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 which are not available. Constraint (20) handles the TA constraint.

Objective function has three pillars. First, the academic staff wants the exams to finish as early as

---

\(^4\)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.
possible to spare time for grading the exams. Therefore the model schedules the classes (especially the crowded
one) 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
\( e_{idt} \). 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 which allows coding with using different
programming languages. We used Python for coding and Gurobi\(^5\) for solving the proposed MIP model.

![Figure 1. Structure of DSS](image.png)

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 is given in Table 4. These tables are created by using data items editor of the Solver
Studio in Figure 2. They are the parameters that are required for running MIP model. They consist of courses,
classrooms, student groups, desired periods and undesired periods. 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.

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

\(^5\)Gurobi(2019). Gurobi [online]. Website [https://www.gurobi.com/] [accessed 30 July 2019]
Figure 2. Input area–data items editor of the 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. Number 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 3(a), for example, there are 107 pairs who have 1 common student and 338 pairs which 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 3(a), Figure 3(b) and Figure 3(c) give the number of tuples of the common students in three and four comparison lists, respectively.

![Pie charts](image)

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

The instances were run on an Intel® Core™ i5-3470 CPU computer with 4 GB RAM. We have prepared the midterm and the final 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 in Table 5. Since 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 department web site and received feedback from both students and academics to revise our model and DSS.

| Exam     | Rows   | Columns | Integer | Non-zero Entries | Iterations | Solution Time (seconds) |
|----------|--------|---------|---------|-------------------|------------|------------------------|
| Midterm  | 204795 | 8640    | 8640    | 671547            | 42982      | 8.08                   |
| Final    | 268814 | 15840   | 15840   | 975688            | 78562      | 17.28                  |

One motivation of this study is 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 comparison of conflicts in midterm exam schedules prepared manually and by DSS for spring midterm. Exam indicates that number of exam in one term and conflicts shows number of students that have conflicts in their exams. In 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 reduced although the number of examinations increased. In order to have a normalized measure, we divided the number of students with conflict to the number of exams. This ratio shows the number of students that has a conflict per exams. 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 thesis of the fourth year’s 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 slot. 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 (i) the number of exams on consecutive days (1st day, 2nd day, etc.), (ii) the number of students taking exams in the first and second week, (iii) 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 don’t give the number of classes assigned to exams since they are almost the same in every instance. It turns out that the number of exams are distributed evenly except the 3-day schedules where the last day has significantly less number of 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, number of students with conflicting exams are quite high for the exam timetables with 3 days and 4 days. Therefore, these cases has been eliminated since the number of students whose exams overlapping is high. Although the parameter values at 4.5 day exam is 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 exam with 5 slot is dominated. For the remaining cases, number of students at slots 6 and 7 are very close to each other. Moreover, students, TAs and instructors do not want to have 7 slots unless it has to exist to maintain feasibility. The number of students taking exam 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 days 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.

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 on MS Excel sheet. Given the enrolment 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 any two consecutive exams, and no more than three exams.
Table 7. Comparative results for the final exam of spring term in 2018-2019

| Case No | Indicies | Parameters | Number of Exam | No. of Students | Last Slot Counts |
|---------|----------|------------|----------------|----------------|------------------|
|         |          |            | Sim. 2 exam    | Con. 2 exam    | Sim. 3 exam      | Con. 3 exam      | M. than 3 exam | 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 5 | 18 18 12 - - | 2888 - 10 652 | - - |
| 3       | 3 7      | 4 2 2 2 2 2 | 16 20 12 - - | 2888 - 11 615 | - - |
| 4       | 4 5      | 4 3 3 3 3 3 | 13 15 9 11 - | 2165 723 | - - |
| 5       | 4 6      | 1 3 1 3 3 2 | 12 13 12 11 - | 2168 720 | 9 635 |
| 6       | 4 7      | 1 2 1 2 2 2 | 11 14 11 12 - | 2092 796 | 11 538 |
| 7       | 4,5 5    | 2 3 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 |

in a day. Since all the required information is defined in DSS, a system-oriented structure has been created by preventing loss of information. Thus, it is always possible to create schedules in certain standards to 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 ETP of other departments and institutions. Anyone with a 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 Vocational High School of Istanbul University which has 7 departments and more than 2000 students. Moreover 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 with YTU-IE.

There are two future directions to be explored. First, the current version uses a 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, fall midterm schedule of Vocational High School of Istanbul University is being prepared by a modified version of our DSS. It takes 30 minutes to find a schedule. Therefore our aim is to use a heuristic method (for example two stage method of [17]) or develop a new method to solve the problem fast. Second, we want to develop a web site so that the DSS can be used on any computer without any required installation.

References

[1] Gashgari R, Alhashimi L, Aljawi L, and 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 O.T, Akinwale A.T, and Aweda O.M. 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 E.K, McCollum B, Merlot L.T.G., and Lee S.Y. 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, and Hadidi A. A survey of approaches for university course timetabling problem. Computers & Industrial Engineering 2015; 86: 43–49.
[5] McCollum B, McMullan P, Parkes A.J, Burke E.K, and Qu R. A new model for automated examination timetabling. Annals of Operations Research 2012; 194(1): 291–315.

[6] Arbaoui T, Boufflet J.P, and 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, and Housos E. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. Annals of Operations Research 2012; 194(1): 203–2221.

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

[9] Cavdur F and 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 A.R and Koupaei M.N. A new binary model for university examination timetabling: a case study. Journal of Industrial Engineering International 2012; 8(1): 28.

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

[12] Qu R and Burke E.K. 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] Cavdur F and 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.

[14] Tilahun S.L. 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, and 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, and Zorn E. Examination timetabling with Moses: System demonstration. In Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT’16); Udine, Italy; 2016. pp. 511–514.

[17] Keskin M.E, Doyen A, Akyer H, and Guler M.G. 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 P.A, and Robles J.M. 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 M.K, and 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 M.D and 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 A.W., Raza S.A., and Tariq Z.M. A web-based group decision support system for academic term preparation. Decision Support Systems, 114:1–17, 2018.