International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

COTTAPP: An Online University Timetable Application based on a Goal Programming Model

Tuğçe Dursun[1], Yasemin Sü[1], Rana Coşgun[1], Ayşe Sevde Durak[1], Barbaros Yet*[1]

Accepted : 13/07/2017 Published: 30/09/2017

Abstract: Preparing university course timetables is a challenging task as many constraints and requirements from the university and lecturers must be satisfied without overlapping courses for different student groups. Although many mathematical optimization models have been proposed to automate this task, a wider use of these models have been limited as deep technical understanding of mathematical and computer programming are required in order to use and implement them. This paper proposes a simple and flexible course timetabling application that is based on a weighted binary goal programming model with a powerful solver. Our application enables the users to modify and run this model by using a simple web and spreadsheet interface. Consequently, the model does not require deep technical understanding of the underlying models from its users even though it is based on a complex mathematical model. The web application and the underlying optimization model is illustrated by using a case study of an undergraduate program of industrial engineering.

Keywords:*University Course Timetabling, Goal Programming, Web Application.*

1. Introduction

A course timetable must be set up at the beginning of every semester in all universities. This challenging task involves assigning courses to limited amount of resources, satisfying the preferences and constraints of lecturers, and preventing overlapping of courses. Therefore, a large amount of time and resources are required when course timetables are manually prepared.

Many mathematical models have been proposed to automate and optimize timetables (see Section 2 for a thorough review). Although, these models can solve this complex problem much faster than humans can, many of them cannot be used by any other people than their developers. This is mainly due to the way these models are presented and published. Previous studies about course timetables published the mathematical details of their model and solution. Although this is sufficient to reproduce the model and implement it to another university, this task requires deep technical expertise about mathematical models and computer programming. Therefore, it limits a wider use and availability of these models. Some studies also published a computer code for their model. However, reading and understanding a computer code, and implementing it to another case study also requires deep technical expertise from a user. As a result, many previous studies have remained as a research project, and their use by other people and institutions have been limited.

In this paper, we present a simple and flexible web application for computing university timetables. Our web application is called the Course Timetabling Application (COTTAPP). COTTAPP solves a complex weighted binary goal programming model by using the IBM CPLEX optimizer [5]. However, it does not require expertise about mathematical modelling or computer programming from the user. Any user who knows how to use spreadsheets and web browsers is able to compute the optimal course timetable by using COTTAPP. The other benefits of COTTAPP include the following:

1. The underlying mathematical model of COTTAPP is flexible: it can incorporate various constraints including preferences and constraints about the course hours and days, and preventing the overlapping of courses.
2. COTTAPP is a free web application that can be run by a simple web-browser. All inputs of COTTAPP can be entered by using an MS Excel spreadsheet, and its outputs can also be exported to spreadsheets.
3. COTTAPP [7] is freely available at "http://ieportal.hacettepe.edu.tr/apps/COTTApp/".

In the remainder of this paper, Section 2 reviews previous mathematical models for university timetabling. Section 3 presents the weighted binary goal programming model that the COTTAPP is based on, and the other features of COTTAPP. Section 4 shows the use of the web application based on a case study, and Section 5 presents our conclusions.

2. Literature Review

One of the earliest models for university course timetabling is based on an integer programming algorithm developed by Akkoyunlu [1]. This model was based on data from a university department, and it was built by the FORTRAN computer programming language. Schniederjans and Kim [2] used binary goal programming model for assignment of instructors to courses according to the departmental requirements and the special requests of instructors. The model could satisfy the departmental requirements, and the requests of instructors were modelled as goals. Dinkel et al. [3] proposed a model with a genetic algorithm based cellular network optimization approach. Main factors like faculty, course, time and physical places were included to the cellular network approach with a penalty function. Classroom, subject and instructor overlaps were not allowed.

[1]*Hacettepe University, Industrial Engineering Department, Ankara/Turkey*
** Corresponding Author:Email: barbaros.yet@hacettepe.edu.tr*

Table 1. Methods and Locations of Timetabling Optimization Studies

Year	*Authors*	*Location*	*Method*
1973	Akkoyunlu[1]	American universities	Binary integer programming
1987	Schniederjans&Kim[2]	University of Nebraska	Binary goal programming
1989	Dinkel et al. [3]	Texas A&M University	Genetic algorithm
1990	Dowsland[4]	University Collage Swansea	Graph coloring, Set partitioning, Simulated annealing
1994	Costa[6]	École Polytechnique Fédérale de Lausanne	Tabu search algorithm
1998	Badri et al. [8].	United Arab Emirates University	Binary goal programming
1998	Colorni et al. [9]	Italian high school	Simulated annealing, Tabu search, Genetic algorithm
1999	Abramson et al. [10]	Griffith University	Simulated Annealing
2000	Deris et al. [11]	Malaysia University	Constraint based reasoning technique
2002	Burke &Petrovic[12]	University of Nottingham	Hyper heuristic methods, multi-criteria, case based approaches
2003	Smith[13]	Monash University	Hopfield neural network
2004	Benli&Botsali[14]	Bilkent University	Binary integer programming
2004	Carrasco&Pato[15]	University of Algarve & University of Lisbon	Potts mean-field annealing simulation, Discrete neural network
2004	Daskalaki&Birbas[16]	Greek universities	Binary integer programming
2005	Daskalaki&Birbas[17]	University of Patras	Two-stage relaxation approach, Binary integer programming
2006	Al-Yakoob&Sherali[18]	Kuwait University	Mixed integer programming
2006	MirHassani[19]	Shahrood Technical University	Binary integer programming
2008	Bakır&Aksop[20]	Gazi University	Binary integer programming
2011	Hao&Benlic[21]	Universitéd'Angers	Binary integer programming, Tabu Search
2017	Dursun et al.	Hacettepe University	Weighted binary goal programming with online interface

Dowsland [4] developed a model for solving multi-objective university timetabling problem at the Management Science Department at University College of Swansea. Dowsland [4] selected a primary objective among the multiple objectives, and used other objectives as constraints for the primary objective. Dowsland [4] used graph coloring, set partitioning and simulated annealing methods to solve this model. Costa [6] used tabu-search heuristic techniques for solving the timetabling problem. Similarly, Colorni et al.[9] used metaheuristic approaches, including simulated annealing, tabu-search and two genetic algorithms, for solving the timetabling problem in an Italian high school and compared the results of these approaches. Abramson et al. [10] compared the performance of six different simulated annealing cooling schedules for the course-timetabling problem.

Badri et al. [8] built a binary goal programming model to prepare course timetables for the United Arab Emirates University. Their model was an extension of Schniederjans and Kim's model [2]. Deris et al. [11] used the constraint based reasoning approach for solving the university course timetabling problem. Their model takes the requirements of the part-time instructors and the special circumstances such as public holidays into account.

Burke and Petrovic [12] focused on hyper-heuristic, multi-criteria and cased-based approaches to prepare a schedule for courses and exams in the University of Nottingham. Smith et al. [13] examined the use of neural networks, greedy simulated annealing and tabu search for the timetabling problem. Carrasco and Pato [15] also use neural networks to prepare timetables for both lecturers and classes.

Daskalaki et al. [16] proposed a model based on binary integer programming to prepare university timetables at Greek Universities. Their model ensures that the courses in the same day are done consequently as blocks. They consider the physical limitations, instructors, courses, days, time intervals and student groups when computing the optimal timetable. Although this increases the complexity of the model it also provides flexibility in implementing the requirements of the institution. Daskalaki and Birbas [17] offered a two-stage relaxation approach to solve such complex models. Their approach is run for every day of a week and aims to find local optimal solutions. Al-Yakoob and Sherali [18] and MirHassani [19] also proposed mixed integer models to prepare timetable for universities in Kuwait and Iran respectively. Hao and Benlic [21] suggested a linear integer model that uses tabu-search techniques and similar to approaches that dividing main problem to sub-problems.

There have been several studies that developed a timetable model for the universities in Turkey. Benli and Botsali [14] used binary integer programming for preparing a timetable for Bilkent University. Their approach is composed of three stages: assigning class meetings, preparing daily course timetables, and assigning classrooms to timetables. Bakır and Aksop [20] proposed a binary integer programming model for Gazi University. Their model incorporates some complex requirements specific to Gazi University. For example, there were different procedures for faculty and non-faculty instructors, and the model aims to satisfy these requirements when computing the optimal timetable.

Table 1 shows the locations and methods used in previous studies and in our study. The majority of these studies, including our study, used a binary integer or goal programming method to compute university timetables. However, the use of many reviewed models have been limited as deep technical knowledge is required to modify and apply these models to some other institution. COTTAPP provides unique benefits in this regard as it can be used by someone who practically has no knowledge about optimization and computer programming. In the following sections, we describe the mathematical model underlying COTTAPP and the web application based on this model.

3. Mathematical Model

This section describes the underlying weighted binary goal programming model that computes timetables in COTTAPP. In the remainder of this section, the requirements of the COTTAPP's model (Section 3.1) and its mathematical formulation (Section 3.2) are discussed.

3.1. Requirements

Our first step was to elicit a list of the requirements expected from a university course timetabling optimization model. For this purpose, we made interviews with two research assistants (RAs)

who are responsible for creating course timetables at the Department of Industrial Engineering in Hacettepe University (HUIE). Our interviews primarily focused on the timetables for the undergraduate degree program at HUIE as this program has the highest number of courses and students.

The basic logical requirements of our model is defined as follows:

- Courses given by the same instructor must not overlap,
- The 'must' courses for a year of study must not overlap,
- All course sessions must be assigned to a time slot,
- If a course has more than 1 lecture hours in a day, those hours must be consecutive.
- There are basic 'service' courses, such as basic mathematics and physics, taught by other departments. Since HUIE students attend these course with students from different departments, these courses' days and hours are fixed and cannot be modified. The model should assign these courses to predefined days and hours.

The RAs indicated that the courses must be grouped for each year of study. For example, each of freshman, sophomore and junior course groups have mandatory courses that must not overlap with other must courses that belong to the group. Similarly, there are 'technical elective' courses for third and fourth year students. These courses should also be defined as a course group as they preferably should not overlap with each other and with 'must' courses of third and fourth year students. Therefore, we defined multiple 'course groups' in our model each representing either courses for a year of study or elective courses.

Satisfying the goals and constraints of the lecturers is a major challenge for preparing a course timetable. Many lecturers have specific preferences about the days and hours they would like to teach. Moreover, some lecturers are not available in some days and hours due to other liabilities and duties. These goals and constraints should be included as soft and hard-constraints as shown below:

- Some courses must be given in a certain day or hour, and this should be included as a hard-constraint.
- Some courses cannot be given in a certain day or hour, and this should be included as a hard-constraint.
- Some courses are preferred to be taught in a certain day or hour, and this should be included as a soft-constraint (i.e. a goal).
- Some courses are not preferred to be taught in a certain day or hour, and this should be included as a soft-constraint.

3.2. Model

We used a weighted binary goal programming approach to build a mathematical model satisfying the requirements and goals discussed in the previous section. The main parameters of the model are shown below:

- *I*: The set of weekdays,
- *J*: The set of possible course periods in a day,
- *M*: The set of courses,
- *K*: The set of instructors who teach more than one course,
- M_t: The set of courses for student group *t,*
- CI_k: The set of courses assigned to instructor *k,*
- *CO*: The set of pairs of courses that must not overlap,
- *CMGP*: The set of courses that must be given in a predetermined day and period,
- *CSGP*: The set of courses that should be given in a predetermined day and period,
- *CMNP*: The set of courses that cannot be given in a predetermined day and period,
- *CMSP*: The set of courses that should not be given in a predetermined day and period,
- *CMGD*: The set of courses that must be given in a predetermined day,
- *CSGD*: The set of courses that should be given in a predetermined day,
- *CMND*: The set of courses that cannot be given in a predetermined day,
- *CSND*: The set of courses that should not be given in a predetermined day,
- *CP(m)*: The total number of periods of a course.
- *IP(i,m)*: The number of course periods assigned to course m in day *i*.
- S_a: The set of soft constraints *a*
- p_a: The coefficient of soft constraints *a*

The decision variables of our model are:

$$x_{ijm} = \begin{cases} 1 \text{ if course } m \text{ is assigned to day } i \text{ and period } j \\ 0 \text{ otherwise} \end{cases}$$

where $\forall_i \in I, \forall_j \in J$ and $\forall_m \in M$,

$d_{s_1}^+$: The positive deviation from non-overlapping technical elective courses, $\forall_{s_1} \in S_1$,

$d_{s_2}^+$: The positive deviation from the preferred number of courses assigned in a day or hour, $\forall_{s_2} \in S_2$,

$d_{s_3}^+$: The positive deviation from the course groups that are not preferred to be overlapped, $\forall_{s_3} \in S_3$,

$d_{s_4}^-$: The negative deviation from courses that are preferred to be taught on certain days, $\forall_{s_4} \in S_4$,

$d_{s_5}^+$: The positive deviation from courses that are not preferred to be taught on certain days, $\forall_{s_5} \in S_5$,

$d_{s_6}^-$: The negative deviation from courses that are preferred to be taught on certain days and hours, $\forall_{s_6} \in S_6$,

$d_{s_7}^+$: The positive deviation from courses that are not preferred to be taught on certain days and hours, $\forall_{s_7} \in S_7$.

The objective function of our model is:

$$Minimize\ z = p_1 d_{s_1}^+ + p_2 d_{s_2}^+ + p_3 d_{s_3}^+ + p_4 d_{s_4}^- + p_5 d_{s_5}^+ + p_6 d_{s_6}^- + p_7 d_{s_7}^+$$

Finally, the constraints are as follows:

Constraint 1: Instructor k cannot give more than one course at the same time.

$$\sum_{m \in M_k} x_{ijm} \leq 1, \quad \forall_i \in I, \forall_j \in J, \forall_k \in K$$

Constraint 2: All of the courses must be assigned weekly.

$$\sum_{i \in I} \sum_{j \in J} x_{ijm} = CP(m), \quad \forall_m \in M$$

Constraint 3: The courses in each course group must not overlap.

$$\sum_{m \in M1} x_{ijm} \leq 1, \quad \forall_i \in I, \forall_j \in J$$

$$\sum_{m \in M2} x_{ijm} \leq 1, \quad \forall_i \in I, \forall_j \in J$$

$$\sum_{m\epsilon M3} x_{ijm} \leq 1, \quad \forall_i \epsilon I, \forall_j \epsilon J$$

Constraint 4: Technical elective courses should not overlap. Note that the fourth course group is assigned to technical elective courses in our model.

$$\sum_{m\epsilon M4} x_{ijm} - d^+_{s_1} \leq 1, \quad \forall_i \epsilon I, \forall_j \epsilon J, \forall_{s_1} \epsilon S_1$$

Constraint 5: In order to discourage the model from assigning the majority of courses on a few days and hours, we penalize if more than four courses are assigned to a day or hour. This is a soft constraint as it is inevitable to assign more than four courses if many courses are being taught.

$$\sum_{m\epsilon M} x_{ijm} - d^+_{s_2} \leq 4, \quad \forall_i \epsilon I, \forall_j \epsilon J, \forall_{s_2} \epsilon S_2$$

Constraint 6: Pairs of courses in the set CO must not overlap

$$x_{ijm^a} + x_{ijm^b} \leq 1, \quad \forall_{(m^a, m^b)} \epsilon CO, \forall_i \epsilon I, \forall_j \epsilon J$$

Constraint 7: Courses in some course groups are preferred to not to overlap. In our case study, technical elective courses and the mandatory courses of the third years should not overlap as third year students have to take some technical elective courses.

$$\sum_{m\epsilon M3} x_{ijm} + \sum_{m\epsilon M4} x_{ijm} - d^+_{s_3} \leq 1, \quad \forall_i \epsilon I, \forall_j \epsilon J, \forall_{s_3} \epsilon S_3$$

Constraint 8: If a course m has more than 1 hours of lectures in a day, those hours must be consecutive. We assume that a teaching day is 9 hours.

$$x_{i1m} - x_{itm} \leq 0,$$

$$\forall_i \epsilon I, \forall_m \epsilon M, \forall_t \epsilon \{2, \dots, IP(i,m)\}, IP(i,m) > 1;$$

$$-x_{ijm} + x_{i(j+1)m} - x_{i(j+t)m} \leq 0,$$

$$\forall_i \epsilon I, \forall_m \epsilon M, \forall_j \epsilon J, \forall_t \epsilon \{2, \dots, IP(i,m)\}, IP(i,m) > 1,$$

$$j + t \leq 9;$$

$$x_{i9m} - x_{i(9-t)m} \leq 0,$$

$$\forall_i \epsilon I, \forall_m \epsilon M, \forall_t \epsilon \{1, \dots, IP(i,m)\}, IP(i,m) > 1.$$

The remainder of the constraints represent the lecturer preferences (goals) and constraints about teaching a course at particular days and hours.

Constraint 9: Some courses cannot be taught on certain days.

$$\sum_{j\epsilon J} x_{ijm} = 0, \quad \forall_{(i,m)} \epsilon CMND$$

Constraint 10: Some courses have to be taught on certain days.

$$\sum_{j\epsilon J} x_{ijm} = CP(m), \quad \forall_{(i,m)} \epsilon CMGD$$

Constraint 11: Some courses are preferred to be taught on certain days.

$$\sum_{j\epsilon J} x_{ijm} + d^-_{s_4} = CP(m), \quad \forall_{(i,m)} \epsilon CSGD, \forall_{s_4} \epsilon S_4$$

Constraint 12: Some courses are not preferred to be taught on certain days.

$$\sum_{j\epsilon J} x_{ijm} - d^+_{s_5} = 0, \quad \forall_{(i,m)} \epsilon CSND, \forall_{s_5} \epsilon S_5$$

Constraint 13: Some courses must be given at predetermined days and hours. The model cannot change the time of these courses.

$$x_{ijm} = 1, \quad \forall_{(i,j,m)} \epsilon CMGP$$

Constraint 14: Some courses cannot be assigned to certain days and hours.

$$x_{ijm} = 0, \quad \forall_{(i,j,m)} \epsilon CMNP$$

Constraint 15: Some courses are preferred to be taught at predetermined days and hours.

$$x_{ijm} + d^-_{s_6} = 1, \quad \forall_{(i,j,m)} \epsilon CSGP, \forall_{s_6} \epsilon S_6$$

Constraint 16: Some courses are not preferred to be taught at certain days and hours.

$$x_{ijm} - d^+_{s_7} = 0, \quad \forall_{(i,j,m)} \epsilon CSNP, \forall_{s_7} \epsilon S_7$$

3.3. Web Application

The model described in the previous section requires numerous inputs in order to be adapted to a particular institution. Our web application COTTAPP [7] provides a graphical and spreadsheet interface to enter the inputs and to run this model. COTTAPP was developed by using R statistical software [22] and Shiny [23]. Shiny is a web interface that creates web applications based on R code. The underlying R code prepares the weighted binary goal programming model as described in Section 3.2, and solves this model by using the IBM CPLEX optimizer. We used an R interface to CPLEX called Rcplex for this task. Figure 1 shows the main interface of COTTAPP.

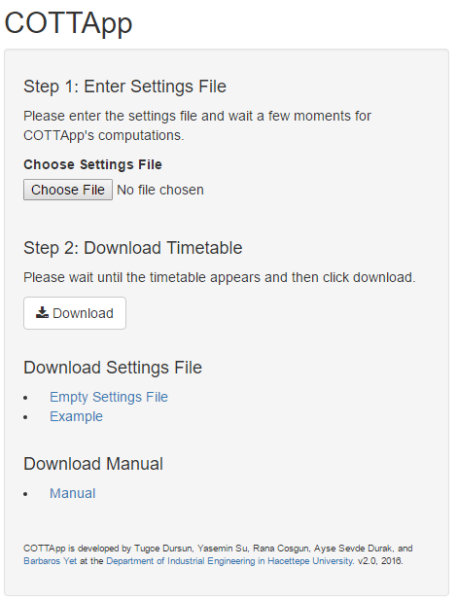


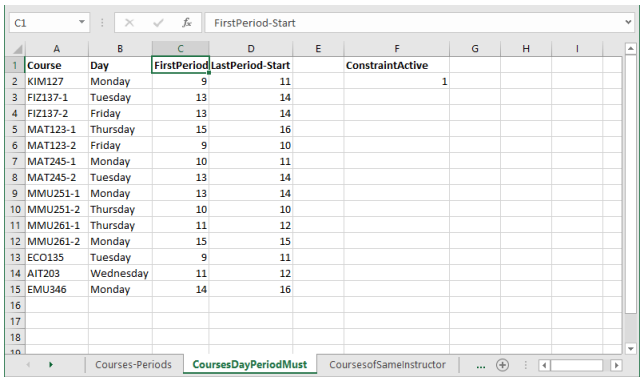
Figure 1. COTTAPP Web Interface

In order to run COTTAPP, a user first needs to download the settings file and enter the required inputs to this spreadsheet. The settings file is a spreadsheet for entering the inputs, and Figure 2 shows a screen shot of the spreadsheet. After the inputs are entered, a user uploads the setting file to COTTAPP, and then the underlying model is automatically run and the results are shown on the web application and downloaded as a MS EXCEL

Table 2. Course Groups, Available Days and Hours (Sheet 1)

Freshman	H/W	Soph.	H/W	Junior	H/W	Tech. Elec.	H/W	CG 5	H/W	CG 6	H/W	Days	Hours
KIM127	3	MAT245-1	2	EMU331	3	EMU434	3	-	0	-	0	Monday	9
ING111-112	3	MAT245-2	2	EMU363	3	EMU346	3					Tuesday	10
FIZ137-1	2	MMU251-1	2	EMU321	3	EMU432	3					Wednesday	11
FIZ137-2	2	MMU251-2	1	EMU341	3	EMU461	3					Thursday	12
EMU111	3	MMU261-1	2			EMU436	3					Friday	13
EMU101	3	MMU261-2	1			EMU451	3						14
MAT123-1	2	ECO135	3										15
MAT123-2	2	AIT203	2										16
		EMU231	3										17

spreadsheet. In the following section, we illustrate the use of COTTAPP and the settings file based on a real case study in HUIE.



	A	B	C	D	E	F	G	H	I
1	Course	Day	FirstPeriod	LastPeriod-Start		ConstraintActive			
2	KIM127	Monday	9	11		1			
3	FIZ137-1	Tuesday	13	14					
4	FIZ137-2	Friday	13	14					
5	MAT123-1	Thursday	15	16					
6	MAT123-2	Friday	9	10					
7	MAT245-1	Monday	10	11					
8	MAT245-2	Tuesday	13	14					
9	MMU251-1	Monday	13	14					
10	MMU251-2	Thursday	10	10					
11	MMU261-1	Thursday	11	12					
12	MMU261-2	Monday	15	15					
13	ECO135	Tuesday	9	11					
14	AIT203	Wednesday	11	12					
15	EMU346	Monday	14	16					
16									
17									
18									

Figure 2. COTTAPP Input Spreadsheet

4. Case Study

In this section, we illustrate the use of COTTAPP by preparing a timetable for the undergraduate degree program of HUIE in the Fall 2015/2016 semester. HUIE has a 4-year undergraduate program in which freshman, sophomore and junior students have to take several mandatory courses. Junior and senior students must also complete a number of 'technical elective' courses which they can select from a list of available technical elective courses. Each lecture takes 50 minutes followed by a 10-minute break. All lecture hours of the courses that take 2 or 3 hours per week are assigned to the same day. Course hours of longer courses are divided to different days according to the preference of the lecturer.

We will use the spreadsheet interface of COTTAPP to enter the inputs of HUIE. An empty spreadsheet interface or an example can be downloaded from the COTTAPP website.

The first sheet in the COTTAPP input spreadsheet is called 'Courses-Periods' (see Table 2). We define the course groups and available lecture days and hours in this sheet. COTTAPP is able to prepare a timetable for up to 6 course groups. In this case study, we only need 4 course groups for freshman, sophomore and junior must courses, and technical elective courses. Therefore, course groups 5 and 6 (CG 5 and CG 6) are left empty in the spreadsheet. The last two columns define the names of the weekdays and available lecture hours each day for the model.

The second sheet, called 'CoursesDayPeriodMust', defines the courses that must be assigned to a predetermined day and period in the timetable (see Table 3). Usually, the times of service courses, such as basic physics and calculus, are defined in this sheet. The course code, the day and starting hours of the first and last period of the course is written to the columns 1 – 4 respectively. For example, the first row of this sheet defines that the KIM127 course must be assigned to the three-hour slot on Mondays between 9:00 – 12:00. The cell below 'ConstraintActive' defines whether this constraint will be active in the mathematical model. If the user enters zero value to that cell, the model will not take the inputs in this sheet into account.

Table 3. Courses with Fixed Days and Hours (Sheet 2)

Course	Day	First Period-Start	Last Period-Start	Constraint Active
KIM127	Monday	9	11	1
FIZ137-1	Tuesday	13	14	
FIZ137-2	Friday	13	14	
MAT123-1	Thursday	15	16	
MAT123-2	Friday	9	10	
MAT245-1	Monday	10	11	
MAT245-2	Tuesday	13	14	
MMU251-1	Monday	13	14	
MMU251-2	Thursday	10	10	
MMU261-1	Thursday	11	12	
MMU261-2	Monday	15	15	
ECO135	Tuesday	9	11	
AIT203	Wednesday	11	12	
EMU346	Monday	14	16	

Table 4. Courses Taught by the Same Lecturer (Sheet 3)

Course1	Course2	Course3	Course4	Course5	Constraint Active
EMU111	EMU363				1
EMU331	EMU432				
EMU434	EMU436				
EMU461	EMU451				

The courses that are given by the same lecturer are defined in the third sheet (see Table 4). Each row of this sheet represents a lecturer who teaches more than one course in the semester. For example, the first lecturer teaches both EMU111 and EMU363 this semester.

The fourth sheet offers another interface for the courses that must not overlap. In this sheet, pairs of courses that must not overlap are defined at each row. A different instructor teaches EMU101 in HUIE, therefore it must not overlap with all other courses taught by those instructors. Table 5 shows how the necessary inputs for this constraint are entered to the fourth sheet of the spreadsheet. Other pairs of courses that must not clash can also be added to this sheet.

Tables 2 – 5 defines the basic inputs required for building a timetable for the HUIE's undergraduate course. However, the

main challenge of preparing a timetable is to incorporate the goals, preferences and requests of the lecturers to the model. For example, some lecturers may not prefer to teach on a particular day, others may have duties that prevents them to teach at certain hours. These preferences and constraints are entered to sheets 5 – 10 in the spreadsheet.

Table 5. Pairs of Courses that Must not Overlap (Sheet 4)

Course1	Course2	ConstraintActive
EMU101	EMU111	1
EMU101	EMU231	
EMU101	EMU221	
EMU101	EMU331	
EMU101	EMU363	
EMU101	EMU321	
EMU101	EMU341	
EMU101	EMU434	
EMU101	EMU432	
EMU101	EMU461	
EMU101	EMU436	
EMU101	EMU451	

If a lecturer prefers to teach a course in a particular day and hours, this input is entered to the fifth sheet in the spreadsheet. Table 6 shows the requests from HUIE. For example, the first row of this sheet represents that EMU434 is preferred to be taught on Fridays between 10:00 – 13:00. The inputs from the fifth sheet is entered as a soft constraint (goal) to the model. The inputs for the courses that cannot or preferred not to be taught on certain days are hours are entered to sheets 6 and 7 respectively in the same way as shown in Table 6.

Table 6. Courses Preferred to be Taught on a Certain Day and Hour (Sheet 5)

Course	Day	FirstPeriod-Start	LastPeriod-Start	Constraint Active
EMU434	Friday	10	12	1
EMU363	Thursday	9	11	

If a lecturer prefers to teach a course in a particular day but the time in that day does not matter, this input is entered to the eight sheet in the spreadsheet. The first row of Table 7 represents that EMU451 must be given on Mondays. Similarly, the inputs for the courses that are preferred to be taught on certain days are entered to sheet 9, and the inputs of the courses that cannot be or preferred not to be taught on certain days are entered to sheets 10 and 11 in the same way.

Table 7. Courses that Must be Taught on a Certain Day (Sheet 8)

Course	Day	ConstraintActive
EMU451	Monday	1

After all the inputs are entered to the spreadsheet, the spreadsheet is uploaded to the COTTAPP website. COTTAPP reads these inputs and runs the model described in Section 3 using these inputs and the CPLEX solver. The results are calculated under 5 seconds, and they are shown in Table 8. The results can also be downloaded in an Excel spreadsheet from COTTAPP.

5. Conclusion

This paper presented an online application, called COTTAPP, to compute optimal university course timetables by using an underlying mixed binary integer optimization. COTTAPP offers simple graphical and spreadsheet interfaces to enter the requirements of the timetable, and then it computes the optimal timetable by solving the optimization model using the IBM CPLEX solver. Although many course timetable optimizers have been developed in previous studies (see Section 2), a wide use of these models have not been possible. This was mainly due to their availability and ease of use. Many of the reviewed studies only presented the mathematical formulation of the model, and the algorithms for solving the model. Therefore, deep technical knowledge is required from potential users in order them to apply it to their domain. The other models were only published as computer code, and thus they still require math and computer skills from the user to understand, modify and implement to code to their case. COTTAPP offers unique benefits in this regard. Although COTTAPP uses a complex model and a powerful solver, it does not require any technical knowledge from the user apart from knowledge to use web browsers and spreadsheet.

As further research, interfaces to implement COTTAPP to the course databases, such as Moodle, could be developed. This would make it possible to automatically input the course names

Table 8. Timetable Created by COTTAPP

	Monday			Tuesday			Wednesday		Thursday			Friday	
09:00	KIM 127		EMU 451		ECO 135	EMU 331		EMU 321			EMU 363	MAT 123-2	
10:00	KIM 127	MAT 245-1	EMU 451	EMU 111	ECO 135	EMU 331		EMU 321		MMU 251-2	EMU 363	MAT 123-2	EMU 434
11:00	KIM 127	MAT 245-1	EMU 451	EMU 111	ECO 135	EMU 331	AIT 203	EMU 321		MMU 261-1	EMU 363		EMU 434
12:00	EMU 101			EMU 111			AIT 203		ING 111-112	MMU 261-1			EMU 434
13:00	EMU 101	MMU 251-1		FIZ 137-1	MAT 245-2				ING 111-112			FIZ 137-2	
14:00	EMU 101	MMU 251-1	EMU 346	FIZ 137-1	MAT 245-2	EMU 436	EMU 221	EMU 341	ING 111-112	EMU 231	EMU 461	FIZ 137-2	EMU 432
15:00		MMU 261-2	EMU 346			EMU 436	EMU 221	EMU 341	MAT 123-1	EMU 231	EMU 461		EMU 432
16:00			EMU 346			EMU 436	EMU 221	EMU 341	MAT 123-1	EMU 231	EMU 461		EMU 432

and groups, and therefore reduce the effort required from the user. We also aim to extend the scope of the COTTAPP's spreadsheet interface to cover any number of student groups and therefore prepare larger tables. The spreadsheet interface could also be incorporated to the web applications graphical interface to provide a more compact input interface for the users.

Acknowledgements

We acknowledge useful discussions with Dr. Volkan Sönmez and Yeşim Koca.

References

1. Akkoyunlu, E.A., A linear algorithm for computing the optimum university timetable. The Computer Journal, 1973. **16**(4): p. 347-350.
2. Schniederjans, M.J. and G.C. Kim, A goal programming model to optimize departmental preference in course assignments. Computers & Operations Research, 1987. **14**(2): p. 87-96.
3. Dinkel, J.J., J. Mote, and M. Venkataramanan, Or practice—An efficient decision support system for academic course scheduling. Operations Research, 1989. **37**(6): p. 853-864.
4. Dowsland, K.A., A timetabling problem in which clashes are inevitable. Journal of the Operational Research Society, 1990. **41**(10): p. 907-918.
5. IBM. IBM CPLEX Optimizer. 2017 03/02/2017]; Available from: http://www.ibm.com/software/commerce/optimization/cplex-optimizer/.
6. Costa, D., A tabu search algorithm for computing an operational timetable. European Journal of Operational Research, 1994. **76**(1): p. 98-110.
7. COTTAPP. COTTAPP: Course Timetabling Applicatoin. 2017; Available from: http://ieportal.hacettepe.edu.tr/apps/COTTApp/.
8. Badri, M.A., et al., A multi-objective course scheduling model: Combining faculty preferences for courses and times. Computers & operations research, 1998. **25**(4): p. 303-316.
9. Colorni, A., M. Dorigo, and V. Maniezzo, Metaheuristics for high school timetabling. Computational optimization and applications, 1998. **9**(3): p. 275-298.
10. Abramson, D., M.K. Amoorthy, and H. Dang, Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific Journal of Operational Research, 1999. **16**(1): p. 1.
11. Deris, S., S. Omatu, and H. Ohta, Timetable planning using the constraint-based reasoning. Computers & Operations Research, 2000. **27**(9): p. 819-840.
12. Burke, E.K. and S. Petrovic, Recent research directions in automated timetabling. European Journal of Operational Research, 2002. **140**(2): p. 266-280.
13. Smith, K.A., D. Abramson, and D. Duke, Hopfield neural networks for timetabling: formulations, methods, and comparative results. Computers & industrial engineering, 2003. **44**(2): p. 283-305.
14. Benli, O.S. and A. Botsali, An optimization-based decision support system for a university timetabling problem: An integrated constraint and binary integer programming approach. Computers and Industrial Engineering, 2004: p. 1-29.
15. Carrasco, M.P. and M.V. Pato, A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. European Journal of Operational Research, 2004. **153**(1): p. 65-79.
16. Daskalaki, S., T. Birbas, and E. Housos, An integer programming formulation for a case study in university timetabling. European Journal of Operational Research, 2004. **153**(1): p. 117-135.
17. Daskalaki, S. and T. Birbas, Efficient solutions for a university timetabling problem through integer programming. European Journal of Operational Research, 2005. **160**(1): p. 106-120.
18. Al-Yakoob, S.M. and H.D. Sherali, Mathematical programming models and algorithms for a class–faculty assignment problem. European Journal of Operational Research, 2006. **173**(2): p. 488-507.
19. MirHassani, S., A computational approach to enhancing course timetabling with integer programming. Applied Mathematics and Computation, 2006. **175**(1): p. 814-822.
20. Bakır, M.A. and C. Aksop, A 0-1 integer programming approach to a university timetabling problem. Hacettepe Journal of Mathematics and Statistics, 2008. **37**(1): p. 41-55.
21. Hao, J.-K. and U. Benlic, Lower bounds for the ITC-2007 curriculum-based course timetabling problem. European Journal of Operational Research, 2011. **212**(3): p. 464-472.
22. R Development Core Team. R: A language and environment for statistical computing. 2017 03/02/2017]; Available from: http://www.R-project.org.
23. RStudio. Shiny: A web application framework for R. 2017 03/02/2017]; Available from: https://shiny.rstudio.com/.