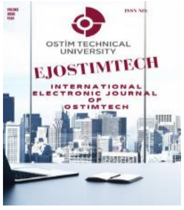

OSTİM TECHNICAL UNIVERSITY

EJOSTIMTECH International Electronic Journal of OSTIMTECH

EJOSTIMTECH, 2022; 1(1):29-34

A MIXED INTEGER PROGRAMMING METHOD FOR UNIVERSITY COURSE SCHEDULING PROBLEM

Serdar Çelik*

Şeyda Ok**

Abstract

It is included in the class of NP-hard problems due to its course scheduling structure and it is encountered in many areas such as university, high school and working life. As the complexity of the problem increases due to the nature of the problem, it becomes increasingly difficult to reach the desired optimal solutions. In the field of education, the scheduling problem is divided into two as lesson scheduling and exam scheduling. Lesson scheduling is also divided into curriculum-based and enrollment-based. On the basis of registration, it is clear which students choose which courses and efforts are made to prevent overlaps. In curriculum-based scheduling, the course information chosen by the students is not known. In this context of obscurity, it is tried to prevent the lessons of the students who take the same course from overlapping. In this study, the course scheduling problems of the students who take curriculum-based courses in the departments of the Faculty of Economics and Administrative Sciences at a foundation university are discussed. Manually solving the problem involves a lot of staff and requirements. Therefore, it is necessary to produce a solution for the problem. Since the problem is unique for the institution, it is necessary to develop a model that will produce a solution specific to the institution. For this purpose, a mixed integer programming model was designed and solved with the help of GAMS program to solve the course scheduling problem.

Keywords: Course Timetabling, University, Mixed Integer Programming

Introduction

Scheduling problems are encountered in many fields, such as education, logistics, and health, and are usually solved by the past experience of people specialized in the field. Scheduling problems, which are combinatorial optimization problems, are NP-hard problems that are difficult to solve by nature. When there are few data and conditions, complete solutions to the problem can be found, but as the data increases, solutions cannot be found in a reasonable time. In scheduling problems, there are often multiple solutions, and it may be considered sufficient to obtain results close to the solution in a reasonable time instead of a single and complete solution (Diveev & Bobr, 2017).

The course scheduling problem is defined as placing courses, classes, and instructors in the same time slot so that they do not overlap under ideal conditions. In the available solution space, a time slot location with no overlap can be considered as a solution. University course scheduling, which is a derivative of scheduling problems, is a difficult optimization problem due to the constraints of classrooms, class hours, and lecturers (Muklason, Irianti, & Marom, 2019). Course scheduling problems in universities and schools typically follow the following algorithmic sequence. First, the issue is handled on a departmental level. That is, each department within the faculty develops its own program, which is then combined. The main disadvantage is that there are not enough resources in the desk area after one department creates its own schedule and passes it to the other department. In this case, the previous schedule must be revised once more. During this time, there is a significant loss of time and workload. As a result, a feasible holistic solution should be obtained within an acceptable time frame.

Scheduling problems, as an optimisation problem, are concerned with the efficient allocation of resources. Many constraints are considered during the scheduling process. Resources are typically limited, and two tasks can be scheduled concurrently at a given time. Artificial intelligence optimisation

* Res. Asst., OSTİM Technical University, Ankara, Turkey, serdar.celik@ostimteknik.edu.tr, ORCID: 0000 0003-4530-9055

** Res. Asst., OSTİM Technical University, Ankara Turkey, seyda.ok@ostimteknik.edu.tr, ORCID: 0000-0002-1157-1701

in solving these problems algorithms such as the Genetic Algorithm, Tabu Search, Ant Colony Optimization, Particle Swarm Optimization, and Artificial Bee Colony are widely used. It is critical to investigate how to implement mix integer programming(mip)-based methods and whether MIP-based methods can achieve the same level of success as other methods in other training scheduling areas.

One of the most important administrative tasks that every university must perform during the academic year is assigning courses to specific time slots and classrooms. Due to the unique characteristics of educational institutions, different studies have emerged and there are many ongoing studies today. The main reason behind this is that the course curricula are constantly updated and new courses are included in the curriculum or courses that are no longer required are excluded from the curriculum.

In this study, an attempt was made to develop a model based on the requests of four departments within the Faculty of Economics and Administrative Sciences. The first section of the study explains who created the model, and the second section explains the model. The model is then applied and solved using the GAMS package program in the following section. The findings and recommendations are presented in the final section.

Literature Review

It is difficult and takes a lot of time for educational institutions to schedule classrooms and capacities, taking into account the needs of instructors and students. Since each institution has its own characteristics, it is not possible to create and implement a standard model. For this reason, the feasibility of the problem attracts the attention of many researchers. (Schaerf, 1999) grouped the course scheduling problem under three headings: course, school-time, exam scheduling. Although they have some common aspects, all three types of problems show significant differences.

School scheduling aims to make a weekly lesson schedule for all classes within the school. Among the important points is the prevention of teachers being assigned to different classes at the same time. At the same time, two teachers are not given to a single class. A detailed study for the problem can be found in the study of (Kingston, 2012).

University course scheduling is a course scheduling that minimizes the overlap of courses with students taking common courses within a faculty or faculty (Botsali, 2000). Exam scheduling is the scheduling of courses with common students by avoiding overlapping of exams and by creating a dispersion of exams across opportunities (Carter, 1986) (Qu et al., 2009). Although there are many intersections between exam and course scheduling, there are also important differences where they differ from each other. For example, in course scheduling, a course is given to a classroom, while in exam scheduling, if the capacity of the classroom is sufficient, more than one course exam can be held in the same classroom. The most obvious difference between course scheduling and school scheduling is that there are students who take common courses at the university. In schools, classes consist of separate students. In other words, if the students of the two courses are common, these courses overlap. They cannot be placed on the same timeline.

University course scheduling class is included in NP-hard problems (Bardadym, 1996). To explain, we have K courses and placing these courses in D classrooms in S time periods means that (S*D)C alternative solutions emerge. As the size of our variables increases, the number of solution candidates will increase.

Many different methods such as heuristics, metaheuristics, and mathematical models have been used in course scheduling problems. (Daskalaki et al., 2004) developed a 0-1 MIP model for the course scheduling problem in their study. Using the goal programming method (Günalay & Şahin, 2006), proposed a solution to the Turkish Military Academy course scheduling problem. (Schimmelpfeng & Helber, 2007) created and implemented a MIP model at Hannover University's Faculty of Economics and Business Administration. They turned this model into a decision-support system and made sure it was used at the university. (Oladokun & Badmus, 2008) created the MIP model in their study and used it in the University of Nigeria, Faculty of Engineering, study for the lessons they organized. (Al-Qaheri, Hasan, & Al-Husain, 2011) at Kuwait University's Faculty of Business Administration, they created a decision support system for course scheduling. Integer goal programming is used to build the model, and assignments are made in three stages: faculty-course, course-time slot, and time slot-class assignment. They tried to minimize this by adding a cost function to the model they developed. (Gunavan et al., 2012) proposed a mathematical model in their study and focused on improving its solution with the help of annealing simulation. (Sánchez-Partida, Martínez-Flores, & Olivares-Benítez, 2014) created a MIP model and solved the UPAEP University course scheduling problem. (Phillips, Waterer, Ehrgott, & Ryan, 2015) investigated the issue of class assignment to courses. For the existing problem, a MIP model with a general structure that can be used for large instances is developed. They proposed a new model that can be solved by the model. (Vermuyten, Lemmens, Marques, & Belien, 2016) proposed a MIP model with two stages. The classrooms and times of the courses were determined in the first stage, and the number of students switching between the courses was minimized in the second stage based on the results of the first stage. (Siddiqui, Raza, & Tariq, 2018) developed a web-based decision support system based on the MIP model they developed for the lesson scheduling problem and implemented it in their universities. They provided course schedule preparation with this study, in which they proposed a new multi-purpose model.

The studies presented above are based on integer models. In this study, the model developed by (Castillo and Alguacil, 2002) for the course scheduling problem was taken into account, and additional constraints were added and rearranged in the model considering the conditions specific to the institution. Two objectives were defined in the model and they were tried to be minimized.

Methodology

Many previous studies have been done and models have been proposed for the course scheduling problem. It has been clearly demonstrated that these models proposed during the implementation phase will yield effective results. Most organizations still try to solve the scheduling problem manually. In this case, too much time and effort are spent to solve the problem, and as a result, a suitable satisfactory solution cannot be found.

In this study, the scheduling situation that a foundation university wants to create for a certain day of the week is discussed. There are 4 departments in total within the Faculty of Economics and Administrative Sciences, and a total of 9 instructors in these departments have the capacity to teach all 30 courses to be given. Instructors consist of faculty members from the faculty or from another institution. Which lecturer will teach which course is taken by the faculty decision. At the same time, these 30 courses were determined as common courses by the four departments. Course hours start at 8 am and end at 18 pm and each course is set to two hours. There is a decision by the faculty administration that each course should be given once and that any department should teach a maximum of 3 courses per hour. Ultimately, an appropriate scheduling solution needs to be devised. For this task, 3 employees are involved in the university and they spend a lot of time. In order to solve the problem in the shortest time frame and to eliminate the loss of workforce, it is important to model the problem effectively and find a solution.

The parameters, constraints and objective function of the course scheduling problem of the four departments of the Faculty of Economics and Administrative Sciences of the foundation university are given below:

Parameters

S_{dr} : The number of classrooms (The representation of the classrooms where the courses will be given is expressed with dr. It will be as Sdr = {classroom1, classroom2, …, classroom6}.)

S_s : The number of course hours (The hours that the courses will be given are expressed in s. The course hours for a day are shown as Ss = {08-10,10-12,12-14,14 16,16-18}.)

S_d : Number of courses (Indicates the number of all courses to be given by the instructors. It is shown as Sd = {course1, course2, …, course30}.

$S_{öd}$: i. Number of courses given by the instructor

S_{bs} : Number of departments (It is the number of departments in the faculty. Sbs = {department1, …department4})

Π : The set of all lessons to be given (Π = {d1, d2, …, d8})

Πi : i. course set given by the instructor (Π1= {d1, d2, d3}, Π2 = {d4, d5, d6, d7}, Π3 = {d8, d9, d10, d11}, Π4 = {d12, d13, d14, d15, d16}, Π5 = {d17, d18, d19, d20, d21, d22}, Π6 = {d23, d24, d25}, Π7 = {d26, d27}, Π8 = {d28, d29}, Π9 = {d30})

Δb : Course set of section b (course set of section 1 Δ_1 = {d1, d2, …, d10}, the course set of part 2 Δ_2 = {d11, d12, …, d17}, the course set of part 3 Δ_3 = {d1, d19, …, d23}, the course set of part 4 Δ_4 = {d24, d25, …, d30})

Note that $\Pi_1 \cup \Pi_2 \cup \Pi_3 \cup \Pi_4 \cup \Pi_5 \cup \Pi_6 \cup \Pi_7 \cup \Pi_8 = \Pi$ and $\Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4 = \Delta$

Decision Variables

The 1-0 decision variable, which shows which course should be given at which time and in which classroom, is expressed below:

V(*d, dr, s*): 1 if course *d* is given in the *dr* classroom at hour *s*, 0 otherwise.

Constraints

Although some aspects of each university are similar, there are some differences between them. These points can also be assigned to faculty, departments, lecturers, classrooms, and working hours. As a result, presenting a comprehensive integer linear programming model that includes every university is impossible. The model to be installed should be custom-made for the institution. Because of the inadequacy of classrooms and a lack of instructors, complexity emerges in our problem. The constraints that meet all the needs of the desired course scheduling problem of the Faculty of Economics and Administrative Sciences are given below:

In constraint (1), each instructor teaches all courses.

$$\sum_{d \in \pi_i} \sum_{dr=1}^{S_{dr}} \sum_{s=1}^{S_s} V(d, dr, s) = S_{öd} \qquad \forall i \tag{1}$$

In constraint (2), each instructor teaches at most 1 course in a time period.

$$\sum_{s \in \pi_i} \sum_{dr=1}^{S_{dr}} V(d, dr, s) \leq 1 \qquad \forall s, \forall i \tag{2}$$

In constraint (3), each course is given once.

$$\sum_{dr=1}^{S_{dr}} \sum_{s=1}^{S_s} V(d, dr, s) = 1 \qquad \forall d \tag{3}$$

In constraint (4), a maximum of 1 course is taught in each class-hour combination.

$$\sum_{s \in \pi} V(d, dr, s) \leq 1 \qquad \forall dr, \forall s \tag{4}$$

In constraint (5), a maximum of 3 courses of any department are taught every hour.

$$\sum_{d \in \Delta_b} \sum_{dr=1}^{S_{dr}} V(d, dr, s) \leq 3 \qquad \forall s, \forall b \tag{5}$$

Objective Function

The objective function tries to minimize two different objectives. It is focused on minimizing the number of classrooms to be used and course hours.

$$Min\sum_{d\in\pi}\sum_{dr=1}^{S_{dr}}\sum_{s=1}^{S_s}(dr+s)V(d,dr,s)$$

Application

The model presented was solved with the GAMS package program using a PC with 12GBRAM, 2.26 GHz processor. The solution was found in 0.13 second. As a result of the problem solved for 6 classrooms, 5 course hours (between 08-18), 30 courses, 9 instructors and 4 departments, all courses were assigned to fill all time periods and all classrooms for one day. The results of the objective function are shown in Table 1.

Table 1. Course Scheduling Results

	08-10	**10-12**	**12-14**	**14-16**	**16-18**
Classroom1	Course 20	Course 21	Course 15	Course 9	Course 16
Classroom2	Course 13	Course 18	Course 24	Course 28	Course 1
Classroom3	Course 8	Course 2	Course 29	Course 27	Course 7
Classroom4	Course 6	Course 26	Course 3	Course 17	Course 30
Classroom5	Course 22	Course 5	Course 4	Course 12	Course 11
Classroom6	Course 10	Course 14	Course 19	Course 23	Course 25

The table shows which instructor is teaching which course in which classroom and at what time. Table 2 shows, for example, the classrooms and hours of the first lecturer's courses.

Table 2. Program for 1st Instructor

	08-10	**10-12**	**12-14**	**14-16**	**16-18**
Classroom 1	-	-	-	-	-
Classroom 2	-	-	-	-	Course1
Classroom 3	-	Course 2	-	-	-
Classroom 4	-	-	Course 3	-	-
Classroom 5	-	-	-	-	-
Classroom 6	-	-	-	-	-

Conclusion

The course scheduling problem encountered in educational institutions is a difficult problem to be overcome at the beginning of the education period for every university. One of the most difficult aspects of the problem is that the curriculum is constantly updated and the optimum course scheduling table found in the previous year loses its functionality the next year. Due to the fact that the departments in the FEAS in the university where the study was conducted were located in a single building, many courses were given jointly and the instructors gave courses to more than one department, it was necessary to consider the problem as a single problem for the faculty. It has been decided that the common courses within the faculty will be given a single day of the week and that the classrooms will be equally distributed to 4 departments for each day on the other days. The aim of the study is to schedule 30 courses for this determined single day. In the study, two different objectives were determined and the most appropriate solution of these objectives was found in line with certain constraints. The related study was carried out in line with the requests of university departments. In future studies, it is thought that a better study can be done in line with the information obtained from the students. In future studies, we hope to convert this system into a web-based structure that universities dealing with similar issues can easily access. Following the development of this decision support system, course schedules in accordance with departmental rules and instructor preferences can be obtained in a timely and efficient manner.

References

Al-Qaheri, H., Hasan, M., & Al-Husain, R. (2011). 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*, 3(2), 95-110.

Bardadym, V. (1996). Computer-aided School and University Timetabling:The New Wave. *Practice and Theory of Automated Timetabling* (s. 22-45). içinde

Botsali, A. R. (2000). *A Timetabling Problem: Constraint and Mathematical Approaches.* Ankara: The Institute of Engineering and Sciences of Bilkent University.

Carter, M. (1986). A Survey of Practical Applications of Examination Timetabling. *Operations Research*, 193-202.

Castillo, E., & Alguacil, N. (2002). Pure and Applied Mathematics Series. E. Castillo, & N. Alguacil içinde, *Building and Solving Mathematical Programming Models in Engineering and Science.* New York: John Wiley and Sons, Inc.

Daskalaki,, S., Birbas, T., & Housos, E. (2004). An İnteger Programming Formulation for a Case Study in University Timetabling. *European Journal of Operational Research*, 117-135.

Diveev, A., & Bobr, O. (2017). Variational Genetic Algorithm for NP-Hard Scheduling Problem Solution. *Elsevier Science Publishers B. V.*, 52–58.

Gunavan, A., Ng, K. M., & Poh, K. L. (2012). A Hybridized Lagrangian Relaxation and Simulated Annealing Method for the Course Timetabling Problem. *Computers & Operations Research*, 3074-3088.

Günalay, Y., & Şahin, T. (2006). A decision support system for the university timetabling problem with instructor preferences. *Asian Journal of Information Technology*, 1479-1484.

Kingston, J. H. (2012). Resource Assignment in High School Timetabling. *Annals of*, 241-54.

Muklason, A., Irianti, R., & Marom, A. (2019). Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm. *Procedia Computer Science*, (161), 656-664.

Oladokun, V., & Badmus, S. (2008). An integer linear programming model of a university course timetabling problem. *The Pacific Journal of Science and Technology*, 9(2), 426-431.

Phillips, A., Waterer, H., Ehrgott, M., & Ryan, D. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53, 42-53.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A Survey of Search Methodologies and Automated System Development for Examination Timetabling. *Springer J Sched*, 55-89.

Sánchez-Partida, D., Martínez-Flores, J., & Olivares-Benítez, E. (2014). An integer linear programming model for a university timetabling problem considering time windows and consecutive periods. *Journal of Applied Operational Research*, 6(3), 158-173.

Schaerf, A. (1999). A Survey of Automated Timetabling. *Artificial Intelligence Review*, 87-127.

Schimmelpfeng, K., & Helber, S. (2007). Application of a real-world university-course timetabling model solved by integer programming. *Or Spectrum*, 6(3), 783-803.

Vermuyten, H., Lemmens, S., Marques, I., & Belien, J. (2016). Developing compact course timetables with optimized student flows. *European Journal of Operational Research*, 651-661.