Pamukkale Univ Muh Bilim Derg, 30(1), 17-30, 2024

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

A simulated annealing algorithm for the faculty-level university course timetabling problem

Fakülte seviyesinde üniversite ders çizelgeleme problemi için bir tavlama benzetimi algoritması

Hatice ERDOĞAN AKBULUT[1], Feristah OZÇELİK[2], Tugba SARAÇ[2]*

[1]Department of Industrial Engineering, Faculty of Engineering and Natural Sciences, Antalya Bilim University, Antalya, Turkey.
hatice.erdogan@antalya.edu.tr

[2]Department of Industrial Engineering, Faculty of Engineering and Architecture, Eskisehir Osmangazi University, Eskisehir, Turkey.
fdurmaz@ogu.edu.tr, tsarac@ogu.edu.tr

Received/Geliş Tarihi: 18.12.2021 Revision/Düzeltme Tarihi: 15.03.2023 doi: 10.5505/pajes.2023.00483
Accepted/Kabul Tarihi: 23.03.2023 Research Article/Araştırma Makalesi

Abstract

In this study, faculty-level university course timetabling problem with double major and minor program constraints where classrooms are shared with several faculties is taken into account. This is the first study considering all these constraints together. A goal programming model is proposed to solve the considered problem. Since it is not possible to find a feasible solution for large-size problems with the proposed model in a time limit, a simulated annealing algorithm is developed. The performance of the proposed solution methods is tested by using randomly generated test problems. In addition, a case study is performed at the engineering faculty of a private university. Computational results show the success of the proposed simulated annealing algorithm to solve large-sized problems. An 83% improvement was achieved with the proposed algorithm for the real-life problem.

Keywords: Faculty-level university course timetabling, Simulated annealing, Goal programming, Double major program, Minor program.

Öz

Bu çalışmada, dersliklerin fakülteler arasında paylaşıldığı, çift anadal ve yan dal kısıtlarının olduğu fakülte seviyesinde üniversite ders çizelgeleme problemi ele alınmıştır. Bu çalışma, tüm bu kısıtları bir arada ele alan ilk çalışmadır. Ele alınan problemi çözmek için bir hedef programlama modeli önerilmiştir. Önerilen model ile büyük boyutlu problemler için süre limiti içinde uygun çözüm bulmak mümkün olmadığından, bir tavlama benzetimi algoritması geliştirilmiştir. Önerilen çözüm yöntemlerinin performansı rassal türetilmiş test problemleri kullanılarak sınanmıştır. Ayrıca özel bir üniversitenin mühendislik fakültesinde vaka çalışması yapılmıştır. Deneysel sonuçlar, önerilen tavlama benzetimi algoritmasının büyük boyutlu problemleri çözmedeki başarısını ortaya koymuştur. Gerçek hayat problemi için önerilen algoritma ile %83 oranında iyileşme sağlanmıştır.

Anahtar kelimeler: Fakülte seviyesinde üniversite ders çizelgeleme, Tavlama benzetimi, Hedef programlama, Çift anadal programı, Yandal programı.

1 Introduction

Scheduling is the arrangement of entities (work, people, tools, courses, etc.) by place and time and subject to constraints to achieve a specific goal [1]. Scheduling problems can be seen in many areas such as manufacturing, health, and education. The timetabling problem, which is a type of scheduling problem, is assigning a specified number of events, such as courses, exams, and meetings, to a limited number of periods in a way to satisfy constraints [2]. In educational institutions, frequently encountered timetabling problems occur with the scheduling of courses and exams [3],[4]. Systemic imperatives, preferences of educational institutions, students, instructors, and limited resources such as classrooms, equipment, and instructors add to the challenges of the educational timetabling problem.

The university course timetabling problem (UCTP) is usually classified as curriculum-based or post-enrollment-based. The curriculum-based course timetabling problem occurs when a conflict arises in assigning courses to a student group dependent on a specific curriculum. The post-enrollment-based course timetabling problem occurs when a conflict arises in assigning courses after registration.

In the literature, the course timetabling problem is considered in different ways. In some studies, courses can be assigned to periods or the classroom, while in other studies, the courses are assigned to both the periods and the classroom. In cases where it is not known formerly which instructor will teach the course, the course is assigned to both the instructor and the period and/or classroom.

In this study, we looked at a UCTP that is based on the curriculum and in which courses are assigned to both times and classrooms. The problem is the faculty-level course timetabling problem, where classrooms are shared by various faculties. For this reason, attention has been paid to the availability of the classroom when assigning the courses to the classroom and period. As it is known, all courses in the department's curriculum must be completed in the double major program. In the minor program, certain courses must be taken. For that reason, when considering double major programs, it is generally desirable not to overlap the compulsory courses of the same semester in both programs. However, since the number of courses required to be taken in minor programs is less, it may be possible to avoid conflicts regardless of the semester. In this way, an opportunity is created for minor students to complete the program on time. By considering the

*Corresponding author/Yazışılan Yazar

constraints of the double major and minor programs together, it is hoped that a more suitable course timetable will be developed for both student groups. This study was motivated by the UCTP at the engineering faculty of a private university. To address the problem, a goal programming model was developed. Also, the simulated annealing (SA) algorithm was proposed to solve large-sized problems. The performance of the proposed methods was tested with randomly generated problems and data taken from the engineering faculty of a private university. The results obtained by both methods were compared.

The paper is structured as follows: Section 2 presents a literature review on the faculty-level course timetabling problem in universities. Section 3 introduces the addressed problem and proposes the goal programming model. Section 4 describes the developed SA algorithm. Section 5 presents the experimental results and interpretations. Finally, Section 6 concludes and suggests future research.

2 Literature review

Detailed information on the university course timetabling problem can be found in the literature review studies of Babaei et al. [5] and Altunay and Eren [6]. In this section, we present studies dealing with the faculty-level course timetabling problem.

MirHassani [7] developed an integer programming model for faculty-level course timetabling problem. In making course-time assignments, MirHassani [7] aimed to minimize conflicts by penalizing the less preferred times and the overlapping of courses.

Ismayilova et al. [8] presented a multi-objective 0-1 integer programming model for faculty-level course timetabling problems that took into account the preferences of both the administration and the faculty members.

Cura [9] proposed an SA approach for the course timetabling problem at the Faculty of Business Administration of Istanbul University. The study aimed to maximize the preferences of faculty members by considering their seniority. The developed approach consisted of two stages: searching for courses that could be placed in the same time slot and assigning the courses to the most appropriate time. In the study, besides the SA approach, the problem was also solved with a genetic algorithm and a tabu search algorithm. As a result, Cura [9] obtained better results with SA compared to other algorithms.

Al Tarawneh and Ayob [10] developed a tabu search method using a multi-neighborhood approach to address the course timetabling problem in Kebangsaan Malaysia University Engineering Faculty.

Nguyen et al. [11] proposed a new hybrid algorithm consisting of a combination of harmony search and the bee algorithm for the UCTP in Vietnam. The proposed approach was compared with variable neighborhood search, tabu search, and bee algorithms by taking fourteen datasets from the Faculty of Information Technologies of the Ho Chi Minh City University of Science in Vietnam.

Demir and Celik [12] addressed a curriculum-based course timetabling problem for Atatürk University Engineering Faculty. The integer programming technique was used for the problem to maximize total satisfaction by considering the preferences of the faculty members. The developed mathematical model reached the optimal solution for the problem with only two departments.

Abdelhalim and El Khayat [13] propose a utilization-based genetic algorithm to maximize resource utilization. In the study, the classroom utilization rate was defined to be used in the objective function. This rate consists of the product of the frequency rate, which measures how often a class is used by the total number of hours it is available, and the occupancy rate, which measures how full a class is relative to its total capacity. The study's goal was to achieve classrooms with the highest occupancy and frequency rates while minimizing other soft constraint violations. It has been observed that unnecessary resource use is prevented with their proposed approach.

Borchani et al. [14] addressed the course timetabling problem of the Faculty of Economics and Administrative Sciences of the University of Tunisia. They proposed a variable neighborhood descent approach to solve the problem. Ertane [15] addressed the problem of assigning courses to the day, time interval, and classroom. The study included the constraint of not overlapping the courses taken by successive student groups in different departments for students doing double majors. To solve the problem, the decomposition method is used in the study.

Ozkan [16] worked on a faculty-level UCTP for Hacettepe University's Faculty of Economics and Administrative Sciences and developed an integer linear programming model and two-stage heuristic algorithm to solve the problem. Thepphakorn and Pongcharoen [17] developed the cuckoo search algorithm and tested it on the data of an engineering faculty. The performance of the developed algorithm was compared with that of particle swarm optimization and hybrid particle swarm optimization algorithms.

Alnowaini and Aljomai [18] suggest an automated system to create faculty timetabling using a genetic algorithm and implement the proposed system for three departments of a faculty. The authors suggested that the developed system be expanded to include the entire faculty in the future. Ariyazand et al. [19] presented a multiobjective model to maximize the desirability of professors. The Faculty of Humanities of the Islamic Azad was selected as a case study, and five different metaheuristics (the bat algorithm, cuckoo search, artificial bee colony, firefly algorithm, and genetic algorithm) were used to solve the problem.

In Table 1, the constraints, the objective functions, and the solution methods of the reviewed studies are given. Explanations for all of the abbreviations used in Table 1 are given in Appendices A1, A2, A3, and B. In this study, H1, H2, H3, H4, H6, H7, H8, H11, H12, H14, H15, H16, and H17 hard constraints were considered. Although it is rarely discussed in the literature, the hard constraint H6 was included in this study because of the common use of classrooms with other faculties. The double major program constraint was addressed by Ertane [15] for the first time. Ertane [15] dealt with minimizing the overlap of the courses of consecutive student groups of different departments. In this study, considering that most of the courses of the first-year student groups in the faculty are common, the double major program constraint is handled to minimize the overlap of the compulsory courses of the same student groups in the different departments. Also, we considered the minor program constraint, and all these constraints were considered together for the first time for the UCTP at the faculty level. Finally, the O2 objective function was adopted in line with the literature.

Table 1. Constraints, objective function, and solution methods of the reviewed studies.

Study	Constraints	Objective functions	Solution methods
MirHassani [7]	H1, H2, H5, H7, H8, H9, H19, H20, S2, S3, S14	O2, O6	IP
Ismayilova et al. [8]	H1,H3,H5	O4, O5	MOP, AHP, ANP
Cura [9]	H1, H2, H3, H4, H20	O3	SA
Al Tarawneh and Ayob [10]	H1, H2, H3, H7, H8, H12, S1, S15, S16	O2	TS
Nguyen et al. [11]	H1, H2, H3, H4, H6, H7, H9, H12, H13, H16, H18, S1, S3, S9, S11, S14, S15, S16, S17	O2	HM (HS, BA)
Demir and Celik [12]	H1, H2, H3, H4, H7, H8, H11, H12, H14, S4	O3	IP
Abdelhalim and El Khayat [13]	H1, H2, H10, H12, H14, H16, S1, S15, S16	O1, O2	GA
Borchani et al. [14]	H1, H2, H3, H12, H14, S16	O2	VND
Ertane [15]	H2, H3, H4, H7, H18, S4, S5, S6, S7, S8,S18	O2,O7	ILP
Ozkan [16]	H1, H2, H3, H9, H11, H12, H14, H15, H16, S4, S10, S14	O2	ILP, TS, SA
Thepphakorn and Pongcharoen [17]	H1, H2, H3, H7, H8, H11, H13, H16, S14	O1, O2	CS
Alnowaini and Aljomai [18]	H1,H2,H3,H12,S11,S12,S14,S15,S16,S17	O6	GA
Ariyazand et al. [19]	H1,H2,H3,H4,H11,H12,S12,S14	O3	BTA, CS, ABC, FA, GA
Our study	H1, H2, H3, H4, H6, H7, H8, H11, H12, H14, H15, H16, H17, S1, S4, S6, S18,S19	O2,O7,O8	GP, SA

3 Problem definition and mathematical model

The motivation of this study is to find a solution to the faculty-level course timetabling problem at a private university. The considered engineering faculty consists of five departments. Based on the curriculum at the beginning of each semester, the departments determine the courses to be opened for all student groups, the weekly hours of each course, and the instructors. In addition, there are common courses given to the same student groups in different departments. Since the number of students who have to take these common courses is large, the courses are divided into sections, and it is determined which student group has to take which section.

During the term, students have both compulsory and elective courses; these courses have a minimum of one and a maximum of four weekly course hours. One-hour and two-hour courses are assigned to one day, while three-hour courses are assigned to either one session or two sessions, determined by the instructors' preference. Four-hour courses are assigned in two sessions on two different days. In addition, the first and second-year student groups of a department have only compulsory courses. The third and fourth-year student groups of a department have both compulsory and elective courses.

In the university, different faculties and vocational schools use the buildings in common. While the classrooms are shared, laboratories and studios are assigned to specified departments or faculties. For this reason, courses are assigned according to the availability of classrooms between 9:00 and 20:00 on weekdays. Courses are assigned to classrooms of a suitable type and capacity. While the course timetabling is generated, course assignments are made according to the instructors' availability.

In this study, a goal programming model was developed for solving the faculty-level course timetabling problem. The model consisted of seven goals listed below:

- *G1* : The two or more hours of courses of the student group should not be assigned during the lunch break,
- *G2* : The different two courses of the student group should not be assigned consecutive periods during the lunch break,
- *G3* : The compulsory courses of consecutive student groups in a department should not overlap,
- *G4* : The course hours of the first or second-year student groups should not exceed certain hours per day,
- *G5* : The course hours of the third or fourth-year student groups should not exceed certain hours per day,
- *G6* : No course hour conflicts for double major program students,
- *G7* : No course hour conflicts for minor program students.

Definitions of index sets, parameters, and decision variables used in the model are given below:

Index sets

$C = \{i, p \mid i, p = 1, \dots, I\}$ Set of course indices

$I = \{j \mid j = 1, \dots, J\}$ Set of instructor indices

$DE = \{m, m' \mid m, m' = 1, \dots, M\}$ Set of department indices

$SG = \{n, u \mid n, u = 1, \dots, N\}$ Set of student group indices

$DA = \{l \mid l = 1, \dots, L\}$ Set of day indices

$P = \{k \mid k = 1, \dots, K\}$ Set of period indices

$C = \{q, o \mid q, o = 1, \dots, Q\}$ Set of classroom indices

$S = \{w \mid w = 1, \dots, W\}$ Set of session indices

$CT = \{v \mid v = 1, \dots, V\}$ Set of classroom-type indices

Parameters

- a_i : The weekly hour of course i,
- b_i : The estimated number of students to take course i,
- h_i : The number of sessions for course i
- g_q : The type of classroom q,
- c_q : The capacity of classroom q,
- d_{imn} : 1, if course i belongs to student group n of department m; 0, otherwise,
- as_{im} : 1, if course i is an elective course of department m; 0, otherwise,
- e_{lkq} : 1, if classroom q is available on day l at period k; 0, otherwise,
- f_{ji} : 1, if instructor j teaches course i; 0, otherwise,

r_{jl} : 1, if instructor j is available on day l; 0, otherwise,
s_{iw} : the hours of session w of course i,
sa_{im} : 1, if course i belongs to the minor program of department m; 0, otherwise,
kk_{iv} : 1, if course i can be assigned to classroom type v; 0, otherwise,
MM : A large enough positive number,
T_1 : The target value for the goal that a maximum of one hour of a course may overlap with the lunch break. It is taken as 1
T_2 : The target value for the goal that a maximum of one course is allowed to overlap with the lunch break. It is taken as 1.
T_3 : The target value for the goal that the maximum number of overlapping compulsory course hours for consecutive student groups at a period. It is taken as 1,
T_4 : The target value for the goal that the number of course hours per day for the first or second-year student groups should not exceed 8 hours. It is taken as 8,
T_5 : The target value for the goal that the number of course hours per day for third or fourth-year student groups should not exceed 8 hours. It is taken as 8
T_6 : The target value for the goal that the maximum number of overlapping course hours for double major students at a period. It is taken as 1
T_7 : The target value for the goal that the maximum number of overlapping course hours for minor students at a period. It is taken as 1.

Decision variables

X_{ilkq} 1, if course i is assigned to classroom q on day l at period k; 0, otherwise
y_{iwlq} 1, if session w of course i is assigned to classroom q on day l; 0, otherwise

$d1^-_{mnqiwlk}, d1^+_{mnqiwlk}, d2^-_{mnlqipok}, d2^+_{mnlqipok}, d3^-_{mniuplk}, d3^+_{mniuplk}, d3^+_{mniuplk}, d4^-_{mnl}, d4^+_{mnl}, d5^-_{mnl} d5^+_{mnl}, d6^-_{mm'nlkqo}, d6^+_{mm'nlkqo}, d7^-_{mm'nlkqo}, d7^+_{mm'nlkqo}$: deviation variables

$d4^{++}_{mnl}$: 1, if the value of $d4^+_{mnl}$ is a positive number; 0, otherwise,
$d5^{++}_{mnl}$: 1, if the value of $d5^+_{mnl}$ is a positive number; 0, otherwise.

The constraints and the objective function of the developed mathematical model are given below:

$$\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{q=1}^{Q} X_{ilkq} = a_i \quad \forall\, i \tag{1}$$

$$\sum_{i=1}^{I}\sum_{q=1}^{Q} d_{imn}\, X_{ilkq} \le 1 \quad \forall\, m,n,l,k \tag{2}$$

$$\sum_{i=1}^{I}\sum_{q=1}^{Q} as_{im}\, X_{ilkq} \le 1 \quad \forall\, m,n,l,k \tag{3}$$

$$\sum_{i=1}^{I}\sum_{q=1}^{Q} f_{ji}\, X_{ilkq} \le 1 \quad \forall\, j,l,k \tag{4}$$

$$\sum_{i=1}^{I} X_{ilkq}\, e_{lkq} \le 1 \quad \forall\, l,k,q \tag{5}$$

$$\sum_{i=1}^{I} X_{ilkq} = 0 \quad \forall\, l,k,q|\; e_{lkq} = 0 \tag{6}$$

$$X_{il(k+1)q} - X_{ilkq} - X_{il(k+2)q} \le 0 \quad \forall\; i,l,k,q,w \mid s_{iw} = 2, (k+2) \le K \tag{7}$$

$$-X_{il(k-1)q} + X_{ilkq} - X_{il(k+1)q} \le 0 \quad \forall\; i,l,k,q,w \mid s_{iw} = 3 \tag{8}$$

$$\sum_{k=1}^{K} X_{ilkq} = y_{i1lq}\, s_{i1} \quad \forall i,l,q|\; h_i = 1 \tag{9}$$

$$\sum_{l=1}^{L}\sum_{q=1}^{Q} y_{i1lq} = 1 \quad \forall i|\; h_i = 1 \tag{10}$$

$$\sum_{k=1}^{K} X_{ilkq} = \sum_{w=1}^{W} y_{iwlq}\, s_{iw} \quad \forall\, i,l,q \mid h_i = 2 \tag{11}$$

$$\sum_{w=1}^{W}\sum_{l=1}^{L}\sum_{q=1}^{Q} y_{iwlq} = 2 \quad \forall\, i \mid\; h_i = 2 \tag{12}$$

$$\sum_{w=1}^{W}\sum_{q=1}^{Q} y_{iwlq} \le 1 \quad \forall\, i,l \mid\; h_i = 2 \tag{13}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{q=1}^{Q} f_{ji}\, X_{ilkq} \le\; MM\, r_{jl} \quad \forall\, j,l \tag{14}$$

$$b_i\;\, X_{ilkq} \le c_q \quad \forall\, i,l,k,q \tag{15}$$

$$\sum_{l=1}^{L}\sum_{k=1}^{K} X_{ilkq} = 0 \quad \forall i,q,v|\; kk_{iv} = 1, g_q \ne v \tag{16}$$

$$\left(\sum_{i=1}^{I} d_{imn}\, X_{ilkq} + \sum_{p=1}^{I} as_{pm}\, X_{plko}\right) \le 1 \quad \forall m,n,l,k,q,o|\; n \in \{3,4\}, q \ne o \tag{17}$$

$$X_{il4q} + X_{il5q} + d1^-_{mnqiwlk} - d1^+_{mnqiwlk} = T_1 \quad \forall\, m,n,q,i,w,l,k \mid s_{iw} > 1 \tag{18}$$

$$X_{il4q} + X_{il5q} + d2^-_{mnlqipok} - d2^+_{mnlqipok} = T_2 \quad \forall\, m,n,i,p,l,k,q,o \mid i \ne p, (q = o \;or\; q \ne o), (\,d_{imn} = 1 \;or\; as_{im} = 1), (\,d_{pmn} = 1 \;or\; as_{pm} = 1) \tag{19}$$

$$\left(\sum_{q=1}^{Q} d_{imn}\, X_{ilkq} + \sum_{q=1}^{Q} d_{pmu}\, X_{plkq}\right) + d3^{-}_{mniuplk} - d3^{+}_{mniuplk} = T_3 \qquad (20)$$

$$\forall\, m,n,i,u,p,l,k \mid i \neq p, u = (n+1), d_{imn} = 1, d_{pmu} = 1$$

$$\left(\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{q=1}^{Q} d_{imn}\, X_{ilkq}\right) + d4^{-}_{mnl} - d4^{+}_{mnl} = T_4 \qquad (21)$$

$$\forall\, m,n,l \mid n \in \{1,2\}$$

$$\left(\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{q=1}^{Q} d_{imn}\, X_{ilkq}\right) + \left(\sum_{p=1}^{I}\sum_{k=1}^{K}\sum_{o=1}^{Q} as_{pm}\, X_{plko}\right) + d5^{-}_{mnl} - d5^{+}_{mnl} = T_5 \qquad (22)$$

$$\forall\, m,n,l \mid n \in \{3,4\}$$

$$d4^{+}_{mnl} \leq MM\, d4^{++}_{mnl} \qquad \forall\, m,n,l \mid n \in \{1,2\} \qquad (23)$$

$$d5^{+}_{mnl} \leq MM\, d5^{++}_{mnl} \qquad \forall\, m,n,l \mid n \in \{3,4\} \qquad (24)$$

$$\left(\sum_{i=1}^{I} d_{imn}\, X_{ilkq} + \sum_{p=1}^{I} d_{pm'n}\, X_{plko}\right) + d6^{-}_{mm'nlkqo} - d6^{+}_{mm'nlkqo} = T_6 \qquad (25)$$

$$\forall\, m,m',n,l,k,q,o \mid m \neq m', q \neq o, n > 1$$

$$\left(\sum_{i=1}^{I} d_{imn}\, X_{ilkq} + \sum_{p=1}^{I} sa_{pm'}\, X_{plko}\right) + d7^{-}_{mm'nlkqo} - d7^{+}_{mm'nlkqo} = T_7 \qquad (26)$$

$$\forall\, m,m',n,l,k,q,o \mid m \neq m', q \neq o, n > 1$$

$$X_{ilkq} \in \{0,1\} \qquad \forall\, i,l,k,q$$

$$y_{iwlq} \in \{0,1\} \qquad \forall\, i,w,l,q$$

$$d4^{++}_{mnl}, d5^{++}_{mnl} \in \{0,1\} \qquad \forall\, m,n,l$$

$$d1^{-}_{mnqiwlk}, d1^{+}_{mnqiwlk}, d2^{-}_{mnlqipok}, d2^{+}_{mnlqipok}, d3^{-}_{mniuplk}, d3^{+}_{mniuplk}, d4^{-}_{mnl}, d4^{+}_{mnl}, d5^{-}_{mnl}, d5^{+}_{mnl}, d6^{-}_{mm'nlkqo}, d6^{+}_{mm'nlkqo}, d7^{-}_{mm'nlkqo}, d7^{+}_{mm'nlkqo} \geq 0 \qquad (27)$$

subject to

$$\min f = \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{q=1}^{Q}\sum_{i=1}^{I}\sum_{\substack{w=1\\ s_{iw}>1}}^{W}\sum_{l=1}^{L}\sum_{k=1}^{K} d1^{+}_{mnqiwlk} \Big/ MNL + \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{q=1}^{Q}\sum_{\substack{i=1\\ (d_{imn}=1\\ \text{or}\\ as_{im}=1)}}^{I}\sum_{\substack{p=1\\ p\neq i\\ (d_{pmn}=1\\ \text{or}\\ as_{pm}=1)}}^{I}\sum_{o=1}^{Q}\sum_{k=1}^{K} d2^{+}_{mnlqipok} \Big/ MNL$$
$$+ \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{\substack{u=1\\ u=n+1}}^{N}\sum_{\substack{p=1\\ p\neq i}}^{I}\sum_{l=1}^{L}\sum_{k=1}^{K} d3^{+}_{mniuplk} \Big/ M(N-1)LK + \sum_{m=1}^{M}\sum_{n=1}^{2}\sum_{l=1}^{L} d4^{++}_{mnl} \Big/ MNL$$
$$+ \sum_{m=1}^{M}\sum_{n=3}^{4}\sum_{l=1}^{L} d5^{++}_{mnl} \Big/ MNL + \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{q=1}^{Q}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\sum_{\substack{o=1\\ o\neq q}}^{Q} d6^{+}_{mm'nlkqo} \Big/ \binom{M}{2}(N-1)LK$$
$$+ \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{q=1}^{Q}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\sum_{\substack{o=1\\ o\neq q}}^{Q} d7^{+}_{mm'nlkqo} \Big/ M(N-1)LK$$

Eq. (1) ensures that all courses are assigned the number of hours per week required by the curriculum. Eq. (2) and Eq. (3) guarantee that a student group does not have more than one compulsory or elective course at the same period on any given day. Eq. (4) guarantees that no instructor is assigned to more than one course and classroom on the same day and period. Eq. (5) does not allow a classroom to be assigned more than one course on a suitable day and period. Eq. (6) guarantees that no course can be assigned to the unavailable day and period of a classroom. If a session has two or three hours, Eq. (7) and Eq. (8), respectively, ensure assigning all of these hours consecutively. Eq. (9) and Eq. (10) ensure that the courses with a single session are assigned to a single day. Eq. (11), Eq. (12), and Eq. (13) provide that the courses with two sessions are assigned on two different days. Eq. (14) ensures that the courses of an instructor are assigned to the instructor's available days. Eq. (15) is the capacity constraint of classrooms. Eq. (16) ensures that courses are assigned to the appropriate type of classroom. Eq. (17) ensures that the compulsory courses of the third- and fourth-year student groups of a department do not conflict with the elective courses of this department. Eq. (18) and Eq. (19) ensure that student groups have at least a one-hour gap during the lunch break (12:00-14:00) as much as possible. It is guaranteed by Eq. (20) that, as much as possible, the compulsory courses of consecutive student groups do not overlap. Eq. (21) ensures that, as much as possible, the total compulsory course hours of the first and second-year student groups of a department do not exceed eight hours in a day. Eq. (22) ensures that, as much as possible, the total hours of compulsory and elective courses of the third and fourth-year student groups of a department do not exceed eight hours in a day. Eq. (23) and Eq. (24) are for the calculation of $d4_{mnl}^{++}$ and $d5_{mnl}^{++}$. Eq. (25) ensures that, as much as possible, a double major program student's compulsory courses and the compulsory courses of the same student group of the double major department do not overlap. Eq. (26) ensures that, as much as possible, a minor program student's compulsory courses do not overlap with the minor program's courses. The sign constraints of decision variables are shown in Eq. (27). The objective function of the model is to minimize the total of the undesired deviations. Each goal was normalized by dividing it by its upper bound, and all goals were considered to have equal weight.

4 Proposed SA algorithm

Simulated annealing (SA) is one of the classic metaheuristic algorithms that has been successfully applied in the solution of many problems in the literature, including university course timetabling problems. The advantages of the simulated annealing algorithm are strong local searching ability and short running time [20]. The SA algorithm has been coded in Python. The details of the developed SA algorithm are given in the following subsections.

4.1 Representation of the solution

The solutions are represented as a matrix, where the rows represent the classrooms and the columns represent the periods. A day was divided into eleven periods, from 9:00 to 20:00. Since only the weekdays were considered, a week consists of 55 periods. Cells of the matrix contain information about the assignment of the courses and the availability of the classrooms. The cell is '0' if the classroom is not available and '1' if it is available. If there is a course code in a cell, this course is assigned to the related time and classroom.

	Monday				Tuesday				Wednesday			Thursday			Friday		
	1	2	..	11	12	..	21	22	23	..	33	34	..	44	45	..	55
CR 1	CS 101	CS 101	..	1	1	..	..	0	1	..	0	CS 101	..	0	0	..	1
CR 2	0	..	..	1	1	..	..	1	IE 201	..	1	1	..	1	1	..	1
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
CR Q-1	1	..	..	0	1	..	..	1	0	..	0	1	..	1	0	..	1
CR Q	1	..	..	1	1	..	IE 201	IE 201	1	..	1	0	..	1	1	..	0

Figure 1. Solution representation.

A sample solution matrix is shown in Figure 1. For example, the first session of CS 101 was assigned to classroom CR1 between 9:00 and 11:00 on Monday, and the second session of CS 101 was assigned to classroom CR1 between 9:00 and 10:00 on Thursday.

4.2 Initial solution

An initial solution was generated by prioritizing the course with the largest number of students and the classrooms with the smallest capacity. The pseudocode of the initial solution generation mechanism is shown in Figure 2.

4.3 Neighborhood structure

In this study, the simple move neighborhood and swap neighborhood structures were used together. In a simple move neighborhood structure, a randomly selected course was moved to a randomly selected available classroom and period. The swap neighborhood structure changes the assignment of two randomly selected courses while ensuring feasibility. The neighbor solutions were generated by using one of the structures that were randomly selected.

```
Sort unassigned courses in descending order of the estimated maximum number of students
Sort classrooms in ascending order of their student capacity
for i in sortedCourses:
    for q in sortedClassrooms:
        Find periods as long as the course hour in classrooms that are available and have enough capacity
        if the available classrooms (q_1) and periods (t_i1) were found;
            if the instructor and student groups of the course i were available at periods (t_i1);
                Check whether the course i was given in two sessions
                    if the course i was given in one session;
                        The course i (session(w)=1) was assigned to classroom (q_1) at periods (t_i1)
                            break
                    else;
                        The course i (w=1) was assigned to the classroom (q_1) at periods (t_i1)
                        Find the periods except for the day of the periods (t_i1) as much as the course hour in classrooms that is available and has enough capacity for course i (w=2)
                        if the available classrooms (q_2) and periods (t_i2) are found;
                            if the instructor and student groups of the course i are available at periods (t_i2);
                                The course i (w=2) was assigned to the classroom (q_2) at periods (t_i2)
                                    break
```

Figure 2. Pseudocode of the initial solution generation mechanism.

4.4 Cooling plan and termination criteria

The cooling plan used in this study is given in Eq. (28).

$$\mathrm{T}_{tc+1} = \alpha \times \mathrm{T}_{tc} \qquad (28)$$

In Eq. (28), tc, T_{tc} represent the temperature change counter and temperature value in step tc, respectively. The algorithm was terminated when the temperature decreased to the final temperature or when the total number of steps (N_{tc}) reached the maximum value it could take or when the objective function value reached 0.

4.5 Pseudocode of the developed SA algorithm

The pseudocode of the developed SA algorithm is shown in Figure 3.

In Figure 3, f, f_{new}, T_0, T_{final}, N_{tc}, tc, rn, T_{tc}, $\propto$, ras represents the current objective function value, the neighbor solution's objective function value, the initial temperature, the final temperature, the total number of steps, the temperature change counter, the number of steps at each temperature, temperature value in step tc, the cooling rate, a random number between 0 and 1, respectively.

5 Experimental results and discussion

The proposed mathematical model and SA algorithm were tested by randomly generated test problems and using engineering faculty data from a private university. With the proposed solution approaches, first a sample problem, then test problems, and a real-life problem are solved.

All tests were performed on a PC with an Intel (R) Core (TM) i5 - 035G11CPU @1.19 GHz processor and 8 GB of RAM. The mathematical model was solved by GAMS/Cplex solver. The time limit was set to 3600 seconds in GAMS.

As a result of the preliminary tests, the values of the final temperature (T_{final}), the cooling rate (α), and the number of steps at each temperature (rn) parameters were fixed as 0, 0.05, and 10, respectively, as they gave successful results.

```
Generate an initial solution
Set T_o ∈ {10, 50, 100}, T_final = 0, ∝= 0.05, rn = 10, N_tc ∈ {100,150,200, 250} parameter values tc = 1, rn = 1
Calculate the current objective function value (f)
  if T_tc > 0, tc < N_tc and f>0;
    if rn < F;
      Randomly select a course
      Find the neighbor solutions of this course, create a res list, and append the neighbor solutions in the res list
      if res > 0;
        Randomly select a neighbor solution from the res list
        if the selected neighbor solution was found by using the simple move neighborhood structure;
          Calculate the objective function value of the selected neighbor solution, f_new
          if f_new <f;
            This course was assigned to a new classroom and period
            f = f_new
            rn = rn + 1
          else;
            if ran(0,1) < exp[−(f_new −f)/T_tc]
              This course was assigned to a new classroom and period
              f = f_new
              rn = rn + 1
            else;
              rn = rn + 1
        else if the selected neighbor solution was found by using a swap neighborhood structure;
          Calculate the objective function value of the selected neighbor solution, f_new
          if f_new < f;
            Swap this course with a selected course
            f = f_new
            rn = rn + 1
          else;
            if ran(0,1) < exp[−(f_new −f)/T_tc]
              Swap this course with a selected course
              f = f_new
              rn = rn + 1
            else;
              rn = rn + 1
      else;
        rn = rn + 1
    else;
      tc = tc + 1
      T_tc+1 = α × T_tc
  else if T_tc = 0 or tc = N_tc or f=0;
    break
```

Figure 3. Pseudocode of the developed SA algorithm.

For the initial temperature (T_0), and the total number of steps (N_{tc}) parameters, more than one successful parameter value was obtained as a result of preliminary tests. Since successful results were obtained when values of 10, 50, or 100 for the initial temperature, and 100, 150, 200, or 250 for the total number of steps were used, all problems were solved with all combinations of these parameter values in 10 repetitions and the best results were reported.

5.1 Sample problem

In the sample problem, there are two departments, four student groups in each department, and 35 courses, including compulsory and elective courses. The parameters of the problem were given in Appendices C1, C2, C3, C4, C5, and C6. The optimal solution was obtained in 32.49 seconds, and the optimality gap is zero with the proposed mathematical model. The obtained timetables for Department 1 and Department 2 are shown in Tables 2 and 3, respectively.

The sample problem has also been solved with the proposed SA algorithm. The SA results for different parameter values are given in Table 4. As seen from the table, the optimal solution was obtained with all parameters in a shorter time than GAMS/Cplex.

5.2 Randomly generated test problems

Two test problems, with three departments and 57 courses (problem 1) and with four departments and 77 courses (problem 2), were generated randomly. For both problems, no feasible solution could be obtained with the mathematical model within the time limit. The results of SA are given in Table 5. As seen from the table, successful results with an objective function value near zero were obtained in a reasonable time.

5.3 Real-Life problem

A real-life problem from a private university's engineering faculty was also solved using the proposed solution methods. There were 107 courses in five departments, 53 instructors, and 31 classrooms in the related term. As seen in Table 6, the best result (f = 0.1264) was achieved in 148.64 seconds, and all results were obtained in less than 300 seconds.

Table 2. Course timetabling of Department 1.

Period\Day	Monday		Tuesday		Wednesday		Thursday		Friday
9:00-10:00	IE 341/ I6/ CR1		MATH 201/ I1/ CR5	IE 407/ I16/ CR2	CS 101/ I3/ CR5	IE 351/ I4/ CR4			IE 433/ I16/ CR3
10:00-11:00	IE 403/ I6/ CR3		MATH 201/ I1/ CR5	IE 407/ I16/ CR2	MATH 101/ I1/ CR5			MATH 211R/ I12/ CR4	IE 433/ I16/ CR3
11:00-12:00	IE 403/ I6/ CR3			IE 407/ I16/ CR2	MATH 101/ I1/ CR5		IE 201/ I4/ CR4	MATH 211/ I5/ CR5	IE 433/ I16/ CR3
12:00-13:00							IE 201/ I4/ CR4	MATH 211/ I5/ CR5	
13:00-14:00		IE 361/ I5/ CR3				CS 415/ I5/ CR3	IE 341/ I6/ CR2	IE 351/ I4/ CR1	
14:00-15:00	MATH 101/ I1/ CR5	IE 361/ I5/ CR3			CS 101L1/I14/ CR6	CS 415/ I5/ CR3	IE 341/ I6/ CR2	IE 351/ I4/ CR1	
15:00-16:00	MATH 101/ I1/ CR5	IE 361/ I5/ CR3	IE 371/ I7/ CR1	PHYS 101L2/ I13/ CR7	CS 101L1/ I14/ CR6	IE 491/ I4/ CR1	CS 101/ I3/ CR5	MATH 101R/ I12/ CR5	
16:00-17:00	IE 401/ I7/ CR5		IE 371/ I7/ CR1	PHYS 101L2/ I13/ CR7	PHYS 101/ I2/ CR5	IE 491/ I4/ CR1	CS 101/ I3/ CR5	MATH 101R/ I12/ CR5	
17:00-18:00	IE 401/ I7/ CR5		IE 371/ I7/ CR1	PHYS 101L2/ I13/ CR7	PHYS 101/ I2/ CR5		PHYS 101L1/ I13/ CR7		IE 405/ I16/ CR3
18:00-19:00	IE 401/ I7/ CR5						PHYS 101L1/ I13/ CR7		IE 405/ I16/ CR3
19:00-20:00	MATH 201/ I1/ CR5	CS 415/ I5/ CR2	PHYS 101/ I2/ CR5		MATH 211/ I5/ CR5	IE 403/ I6/ CR3	PHYS 101L1/ I13/ CR7	IE 201/ I4/ CR2	IE 405/ I16/ CR3

Table 3. Course timetabling of Department 2.

Period\Day	Monday		Tuesday		Wednesday		Thursday		Friday	
9:00-10:00	CS 472/ I17/ CR4		MATH 201/ I1/ CR5		CS 101/ I3/ CR5		CS 201/ ÖI/ CR5	CS 431/ I10/ CR4	CS 311/ I10/ CR5	
10:00-11:00	CS 472/ I17/ CR4		MATH 201/ I1/ CR5		MATH 101/ I1/ CR5	CS 361/ I11/ CR4			MATH 211R/ I12/ CR4	
11:00-12:00	CS 472/ I17/ CR4				MATH 101/ I1/ CR5	CS 361/ I11/ CR4			MATH 211/ I5/ CR5	CS 431/ I10/ CR4
12:00-13:00									MATH 211/ I5/ CR5	CS 431/ I10/ CR4
13:00-14:00			CS 201/ I8/ CR4	CS 491/ I10/ CR3	CS 415/ I5/ CR3					
14:00-15:00	MATH 101/ I1/ CR5	CS 441/ I9/ CR2	CS 201/ I8/ CR4	CS 491/ I10/ CR3	CS 415/ I5/ CR3					

Table 3. Continued.

Period\Day	**Monday**		**Tuesday**		**Wednesday**		**Thursday**		**Friday**	
15:00-16:00	MATH 101/ I1/ CR5	CS 441/ I9/ CR2	CS 101 L2 /I14 /CR6				CS 101/ I3/ CR5		MATH 101R/ I12/ CR5	CS 303/ I9/ CR2
16:00-17:00	CS 201L/ I15/ CR6	CS 441/ I9/ CR2	CS 101 L2 /I14 /CR6		PHYS 101/ I2/ CR5		CS 101/ I3/ CR5		MATH 101R/ I12/ CR5	CS 303/ I9/ CR2
17:00-18:00	CS 201L/ I15/ CR6				PHYS 101/ I2/ CR5	CS 492/ I10/ CR3		CS 101 L3/ I14/ CR6	PHYS 101L3/ I13/ CR7	CS 303/ I9/ CR2
18:00-19:00							CS 311/ I10/ CR2	CS 101 L3/ I14/ CR6	PHYS 101L3/ I13/ CR7	
19:00-20:00	MATH 201/ I1/ CR5	CS 415/ I5/ CR2	PHYS 101/ I2/ CR5	CS 361/ I11/ CR2	MATH 211/ I5/ CR5		CS 311/ I10/ CR2		PHYS 101L3/ I13/ CR7	

Table 4. Results of sample problem using SA.

T_0	N_{tc}	f	*CPU (sec.)*
10	100	0	3.70
10	150	0	**1.95**
10	200	0	2.83
10	250	0	2.90
50	100	0	3.27
50	150	0	2.80
50	200	0	2.85
50	250	0	2.67
100	100	0	3.55
100	150	0	2.02
100	200	0	2.39
100	250	0	4.10

Table 5. Test results for the randomly generated test problems using SA.

		Problem 1		*Problem 2*	
T_0	N_{tc}	f	CPU (sec.)	f	CPU (sec.)
10	100	0.014	54.11	0.0893	465.68
10	150	**0.002**	83.84	0.0878	318.34
10	200	0.004	113.09	0.0833	149.11
10	250	0.004	141.52	0.0792	207.11
50	100	0.012	56.15	0.0858	79.88
50	150	0.006	87.60	**0.0717**	114.41
50	200	0.006	116.99	0.0813	146.62
50	250	0.006	624.45	0.0828	183.35
100	100	0.008	56.63	0.1212	70.53
100	150	0.006	85.24	0.0954	109.84
100	200	0.006	114.41	0.0772	149.49
100	250	0.004	140.60	0.0722	184.18

Table 6. Test results for the real-life problem using SA.

T_0	N_{tc}	F	*CPU (sec.)*
10	100	0.1851	217.84
10	150	0.1503	92.13
10	200	0.1618	114.29
10	250	**0.1264**	148.64
50	100	0.1749	55.00
50	150	0.1768	261.29
50	200	0.1573	288.65
50	250	0.1741	141.76
100	100	0.1861	56.61
100	150	0.1560	86.87
100	200	0.1576	109.96
100	250	0.1622	155.87

The deviation values and the objective function value for the current state and SA result are shown in Table 7. In the current state, while the faculty course timetabling was prepared in approximately 2-3 weeks, a better result was achieved in 148.64 seconds with the SA algorithm. As can be seen from the table, in the current state double major (G6) or minor (G7) students have much more course overlap than the solution obtained with the SA algorithm. Another goal is that the compulsory courses of consecutive student groups do not overlap as much as possible. In other words, G3 was seen to have less course overlap with the SA algorithm compared to the current situation. G4 and G5 were almost provided with the developed SA algorithm. Finally, G1 and G2 were achieved with the developed SA algorithm. Considering the objective function values, an 83% improvement was achieved with the proposed SA for the real-life problem.

Table 7. The comparison of the SA algorithm's result with the current state of the real-life problem.

	Current state	*Proposed SA algorithm*
G1	0.20	0
G2	0.10	0
G3	0.082	0.0364
G4	0.05	0
G5	0.05	0.01
G6	0.048	0.020
G7	0.218	0.06
f	0.7490	0.1264

6 Conclusion

In this study, the faculty-level UCTP was addressed. Since most of the classrooms were shared with other faculties, the availability of the classrooms was taken into consideration. The goals listed below were taken into account in this study, and a goal programming model was developed in line with these objectives.

- Having a time gap at noon for student groups,
- No overlapping of the compulsory courses of consecutive student groups of departments,
- Limiting the course hours of student groups to less than or equal to eight hours per day,
- No overlapping of the courses of the student in the double major or minor programs.

A SA algorithm was developed to tackle large-sized problems. The randomly generated test problems and a real-life problem were solved using the proposed mathematical model and the developed SA algorithm using different parameter values. In addition to the solution time advantage, the SA algorithm reached successful solutions within a reasonable time for problems for which the mathematical model failed to find a feasible solution.

In future studies, new constraints may be added to the faculty-level course timetabling problem, such as not assigning a course to the last time of the day and setting the hours per day for student groups to a specified minimum. The size of the problem can be enlarged to include the courses of more than one faculty and department. Hence, a more suitable course timetable will be developed for students who wish to engage in a double major or minor program both within and across faculties. Different heuristics, meta-heuristics, or hybrid algorithms can be used to solve the considered problem.

7 Author contribution statements

In the scope of this study, Hatice ERDOĞAN AKBULUT in the formation of the idea, the design, the assessment of obtained results, the literature review, supplying the data used, and examining the results; Feristah OZÇELIK, in the formation of the idea, the design, examining the results and the spelling and checking the article in terms of content; Tugba SARAÇ the formation of the idea, the design, examining the results and the spelling and checking the article in terms of content were contributed.

8 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared.

There is no conflict of interest with any person/institution in the article prepared.

9 References

[1] Wren A. *Scheduling, Timetabling and Rostering-A special Relationship?*. Editors: Burke EK, Ross P. Practice and Theory of Automated Timetabling, 46-75, Heidelberg, Berlin, Springer, 1996.

[2] Burke EK, Petrovic S, Qu R. "Case-based heuristic selection for timetabling problems". *Journal of Scheduling*, 9(2), 115-132, 2006.

[3] Petrovic S, Burke E. "University timetabling". *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 45, 1-34, 2004.

[4] Schaerf A. "Survey of automated timetabling". *Artificial Intelligence Review*, 13(2), 87-127, 1999.

[5] Babaei H, Karimpour J, Hadi A. "A survey of approaches for university course timetabling problem". *Computers and Industrial Engineering*, 86, 43-59, 2015.

[6] Altunay H, Eren T. "A literature review for course scheduling problem". *Pamukkale University Journal of Engineering Sciences*, 23(1), 55-70, 2017.

[7] Mirhassani SA. "A computational approach to enhancing course timetabling with integer programming". *Applied Mathematics and Computation*, 175(1), 814-822, 2006.

[8] Ismayilova NA, Sağir M, Gasimov RN. "A multiobjective faculty-course-time slot assignment problem with preferences". *Mathematical and Computer Modelling*, 46(7-8), 1017-1029, 2007.

[9] Cura T. "Timetabling of faculty lectures using simulated annealing algorithm". *İstanbul Commerce University Journal of Science*, 6(12), 1-20, 2007.

[10] Al Tarawneh HY, Ayob M. "Using tabu search with multi-neighborhood structures to solve university course timetable UKM case study (faculty of engineering)". *2011 3rd Conference on Data Mining and Optimization (DMO)*, Putrajaya, Malaysia, 28-29 June 2011.

[11] Nguyen K, Nguyen P, Tran N. "A hybrid algorithm of harmony search and bees algorithm for a university course timetabling problem". *International Journal of Computer Science Issues*, 9(1), 12-17, 2012.

[12] Demir Y, Çelik C. "An Integer Programming Approach for Curriculum Based Timetabling Problem Solution". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31(1), 145-159, 2016.

[13] Abdelhalim EA, El Khayat GA. "A utilization-based genetic algorithm for solving the university timetabling problem (UGA)". *Alexandria Engineering Journal*, 55(2), 1395-1409, 2016.
[14] Borchani R, Elloumi A, Masmoudi M. "Variable neighborhood descent search based algorithms for course timetabling problem: Application to a Tunisian University". *Electronic Notes in Discrete Mathematics*, 58, 119-126, 2017.
[15] Ertane E. Course Timetabling with Double Major Constraints. MSc Thesis, Ataturk University, Erzurum, Turkey, 2018.
[16] Özkan A. Solving University Course Timetabling Problems with Integer Linear Programming and Heuristic Approaches. PhD Thesis, Hacettepe University, Ankara, Turkey, 2019.
[17] Thepphakorn T, Pongcharoen P. "Performance improvement strategies on cuckoo search algorithms for solving the university course timetabling problem". *Expert Systems with Applications*, 161, 1-21, 2020.
[18] Alnowaini G, Aljomai AA. "Genetic Algorithm for Solving University Course Timetabling Problem Using Dynamic Chromosomes". *2021 International Conference of Technology, Science and Administration,* Taiz, Yemen, 22-24 March 2021.
[19] Ariyazand A, Soleimani H, Etebari F, Mehdizadeh E. "Improved satisfaction of university faculty by utilizing the Bat metaheuristic algorithm (Case study of the faculty of humanities, Islamic Azad University, Parand Branch)". *Journal of Industrial Engineering and Management Studies,* 9(8), 95-108, 2022.
[20] Long J, Wu S, Han X, Wang Y, Liu L. "Autonomous task planning method for multi-satellite system based on a hybrid genetic algorithm". *Aerospace*, 10(1), 1-18, 2023.

Appendix A

Appendix A1. Explanations for the abbreviations of hard constraints.

Identifier	Explanation
H1	Instructors cannot be assigned to more than one course or classroom at the same period.
H2	Students/student groups cannot be assigned to more than one course or classroom at the same period.
H3	More than one course, instructor, and student group cannot be assigned to a classroom at the same period.
H4	Courses should be assigned to the timetable to fill the number of weekly course hours.
H5	Instructors should be assigned as many courses as they have to teach daily or weekly.
H6	Courses should be assigned to periods when a classroom is available.
H7	The courses given to a single session must be assigned consecutively throughout the course's duration.
H8	Courses assigned to more than one session must be assigned consecutively in the same amount as the course hours specified in each session, provided that they are not on the same day or consecutive days.
H9	If the course has a predetermined instructor, time, or classroom, it should be assigned to the schedule in advance.
H10	Courses should be assigned to specific days and periods.
H11	Courses should be assigned to a technically feasible classroom.
H12	Courses should be assigned to classrooms with enough capacity, or capacity overruns may be allowed up to a specified ratio.
H13	Courses should be assigned according to the availability of the student groups to which they are assigned.
H14	All courses must be assigned to the timetable.
H15	Compulsory courses and elective courses to be taken by a student/student group should not conflict.
H16	Courses should be assigned according to the instructor's preferences.
H17	Courses can only be assigned to the available periods.
H18	Even if a course is divided into more than one session, all sessions should be assigned to the same classroom.
H19	The number of total daily course hours for a student/student group should not exceed the previously specified number of course hours.
H20	Normal education courses and evening education courses can only be taken at certain times.

Appendix A2. Explanations for the abbreviations of soft constraints.

Identifier	Explanation
S1	The number of courses/course hours that the student group will receive daily should not exceed the specified number of courses/course hours as much as possible.
S2	For a two-session course, there should be at least a one-day gap between sessions, if possible.
S3	Student groups' courses should not overlap as much as possible.
S4	Compulsory courses of student groups in consecutive years should not overlap as much as possible.
S5	The classroom must have enough space for the course, or overcrowding may be permitted up to a particular ratio as much as possible.
S6	Courses of students/student groups should not be assigned to lunch break as much as possible.
S7	Courses of consecutive student groups of different departments should not overlap as much as possible.
S8	Departmental courses should not be assigned as much as possible to the other department's building.
S9	Courses of student groups should be assigned according to the day-time slot preferences as much as possible.
S10	A student group's elective courses should not overlap as much as possible.
S11	Courses of a student group should be assigned to as few days as possible.

Appendix A2. Continued.

Identifier	Explanation
S12	All hours of a course should be given in the same classroom as much as possible.
S13	The hours of a course should be assigned as consecutively as possible.
S14	An instructor's preferences, such as a day-time slot, course, and classroom, should be provided as much as possible.
S15	An instructor's daily or weekly course or course hours should not exceed the specified number of courses or course hours as much as possible.
S16	On any given day or block, there should be as few as possible time gaps between the courses that the student group is taking and/or the instructors are teaching.
S17	An instructor's courses should be assigned to as few days as possible.
S18	Double major program students' courses should not overlap as much as possible.
S19	Minor program students' courses should not overlap as much as possible.

Appendix A3. Explanations for the abbreviations of objective functions.

Identifier	Explanation
O1	Maximizing the classroom utilization
O2	Minimizing the total penalty for violations of soft constraints
O3	Maximizing the instructor's preferences
O4	Minimizing the penalty in case the department's/administrative unit's preferences are not met
O5	Minimizing the penalty in case the instructors' preferences are not met
O6	Maximizing the total number of course-classroom-instructor-time or course-instructor-time assignments
O7	Minimizing the double major program student's course overlap
O8	Minimizing the minor program student's course overlap

Appendix B

Appendix B. Explanations for the abbreviations of solution methods.

Identifier	Explanation
ABC	Artificial Bee Colony
AHP	Analytic Hierarchy Process
ANP	Analytic Network Process
BA	Bees Algorithm
BTA	Bat Algorithm
CS	Cuckoo Search
FA	Firefly Algorithm
GA	Genetic Algorithm
GP	Goal Programming
HS	Harmony Search
HM	Hybrid Method
ILP	Integer Linear Programming
IP	Integer Programming
ILS	Iterated Local Search
MOP	Multi-Objective Programming
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SP	Stochastic Programming
TS	Tabu Search
VND	Variable Neighborhood Descent

Appendix C

Appendix C1. Compulsory courses for the sample problem.

Department	Student group	Course code	Weekly course hours	Number of sessions	Instructor	Maximum number of students to take the course	Classroom type
Department 1	1	MATH 101	4	2	I1	80	1
		MATH 101R	2	1	I12	80	1
		PHYS 101	3	2	I2	80	1
		PHYS 101L1	3	1	I13	25	2
		PHYS 101L2	3	1	I13	25	2
		CS 101	3	2	I3	80	1
		CS 101L1	2	1	I14	35	3
	2	MATH 201	3	2	I1	85	1
		IE 201	3	2	I4	40	1
		MATH 211	3	2	I5	85	1
		MATH 211R	1	1	I12	50	1

Appendix C1. Continued.

Department	Student group	Course code	Weekly course hours	Number of sessions	Instructor	Maximum number of students to take the course	Classroom type
Department 1	3	IE 341	3	2	I6	40	1
		IE 351	3	2	I4	40	1
		IE 361	3	1	I5	40	1
		IE 371	3	1	I7	40	1
	4	IE 403	3	2	I6	40	1
		IE 491	2	1	I4	40	1
Department 2	1	MATH 101	4	2	I1	80	1
		MATH 101R	2	1	I12	80	1
		PHYS 101	3	2	I2	80	1
		PHYS 101L3	3	1	I13	25	2
		CS 101	3	2	I3	80	1
		CS 101L2	2	1	I14	35	3
		CS101L3	2	1	I14	35	3
	2	CS 201	3	2	I8	35	1
		CS 201L	2	1	I15	35	1
		MATH 201	3	2	I1	85	1
		MATH 211	3	2	I5	85	1
		MATH 211R	1	1	I12	50	1
	3	CS 303	3	1	I9	40	1
		CS 311	3	2	I10	40	1
		CS 361	3	2	I11	40	1
	4	CS 491	2	1	I10	40	1
		CS 492	1	1	I10	40	1

Appendix C2. Elective courses for the sample problem.

Department	Course code	Weekly course hours	Number of sessions	Instructor	Maximum number of students who can take the course	Classroom type
Department 1	IE 405	3	1	I16	45	1
	IE 407	3	1	I16	45	1
	IE 433	3	1	I16	50	1
	IE 401	3	1	I7	80	1
Department 2	CS 431	3	2	I10	50	1
	CS 472	3	1	I17	50	1
	CS 441	3	1	I9	50	1
	CS 415	3	2	I5	50	1

Appendix C3. Classrooms for the sample problem.

Classroom name	Classroom type	Classroom capacity
CR1	Normal (1)	40
CR2	Normal (1)	52
CR3	Normal (1)	64
CR4	Normal (1)	68
CR5	Normal (1)	144
CR6	Computer laboratory (3)	44
CR7	Physics laboratory (2)	28

Appendix C4. Availability of classrooms based on the period for the sample problem.

Period	CR1	CR2	CR3	CR4	CR5	CR6	CR7	Period	CR1	CR2	CR3	CR4	CR5	CR6	CR7
1	1	0	1	1	0	1	1	29	1	1	1	0	0	1	1
2	1	0	1	1	0	1	1	30	1	0	1	0	1	1	1
3	1	0	1	1	0	1	1	31	0	0	1	0	1	1	1
4	1	0	1	1	0	1	1	32	0	0	1	0	1	1	1
5	1	1	1	1	1	1	1	33	0	0	1	0	1	1	1
6	1	1	1	0	1	1	1	34	1	1	0	1	1	1	1
7	1	1	1	0	1	1	1	35	1	1	0	1	1	1	1
8	0	1	1	1	1	1	1	36	1	1	0	1	1	1	1
9	0	1	1	1	1	1	1	37	1	1	0	1	1	1	1
10	0	1	1	1	1	1	1	38	1	1	0	1	1	1	1
11	0	1	1	1	1	1	1	39	1	1	0	1	1	1	1
12	0	1	1	0	1	1	1	40	0	1	0	1	1	1	1
13	0	1	1	0	1	1	1	41	0	1	0	1	1	1	1
14	0	1	1	0	1	1	1	42	0	1	0	1	1	1	1
15	1	0	1	0	1	1	1	43	0	1	0	1	1	1	1
16	1	0	1	1	0	1	1	44	0	1	0	1	1	1	1
17	1	1	1	1	0	1	1	45	1	0	1	1	1	1	1
18	1	1	1	1	0	1	1	46	1	0	1	1	1	1	1

Appendix C4. Continued.

Period	CR1	CR2	CR3	CR4	CR5	CR6	CR7
19	1	1	1	1	0	1	1
20	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1
26	1	1	1	1	0	1	1
27	1	1	1	0	0	1	1
28	1	1	1	0	0	1	1

Period	CR1	CR2	CR3	CR4	CR5	CR6	CR7
47	1	0	1	1	1	1	1
48	1	0	1	1	1	1	1
49	1	0	1	1	1	0	1
50	1	1	1	0	1	0	1
51	0	1	1	0	1	0	1
52	0	1	1	0	1	0	1
53	1	1	1	0	0	0	1
54	1	1	1	0	0	0	1
55	1	1	1	0	0	0	1

*1: Available. 0: not available.

Appendix C5. Daily availability of instructors for the sample problem.

Instructor	Monday(1)	Tuesday(2)	Wednesday(3)	Thursday(4)	Friday(5)
I1	1	1	1	0	0
I2	1	1	1	1	0
I3	0	1	1	1	1
I4	0	1	1	1	1
I5	1	1	1	0	1
I6	1	1	1	1	0
I7	1	1	0	0	1
I8	0	1	1	1	0
I9	1	1	1	0	1
I10	0	1	1	1	1
I11	0	1	1	0	1
I12	1	1	0	1	1
I13	0	1	1	1	1
I14	1	1	1	1	0
I15	1	1	1	1	1
I16	0	1	0	0	1
I17	1	0	1	1	0

*1: Available. 0: not available.

Appendix C6. Minor courses of departments for the sample problem.

Department	Minor courses
Department 1	IE 201, IE 341, IE 351
Department 2	CS 201, CS 303, CS 311