

## Research Article

# A Novel Optimization Approach for Educational Class Scheduling with considering the Students and Teachers' Preferences

Yu Chen,<sup>1</sup> Mahmonir Bayanati ,<sup>2</sup> Maryam Ebrahimi,<sup>3</sup> and Sadaf Khalijian <sup>4</sup>

<sup>1</sup>Department of Energy and Materials Engineering, Shandong Polytechnic College, Shandong, Jining, China

<sup>2</sup>Faculty of Technology and Industrial Management, Health and Industry Research Centre, West Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>3</sup>Department of Information Technology Management, Islamic Azad University-Electronic Branch, Tehran, Iran

<sup>4</sup>Faculty of Educational Sciences and Psychology, Department of Educational Administration and Human Resources Development, Ferdowsi University of Mashhad, Mashhad, Iran

Correspondence should be addressed to Sadaf Khalijian; [khalijian@um.ac.ir](mailto:khalijian@um.ac.ir)

Received 5 August 2022; Revised 11 September 2022; Accepted 7 October 2022; Published 15 October 2022

Academic Editor: Peiman Ghasemi

Copyright © 2022 Yu Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

University class scheduling problem is one of the most important and complex issues in the academic field. This problem is recognized as one of the NP-HARD issues due to its various limitations. On the contrary, genetic algorithms are commonly used to solve NP-HARD problems, which is one of the decision-making problems and is basically one of the most fundamental classes of complexity. The university course planning includes severe constraints such as classroom, classroom curriculum, and faculty. At the same time, some soft constraints should be considered, such as student and faculty preferences and favorite class time. In this research, as a novel contribution, an integer model for scheduling university classes is presented. In this model, the preferences of professors and students are in accordance with the satisfaction values obtained through questionnaires. Moreover, a genetic algorithm has been developed to solve the model. The results show that the classroom timeline by this algorithm goes well during each run. Moreover, considering an exploratory search for the genetic algorithm can greatly improve the performance of this algorithm.

## 1. Introduction

Curriculum planning and schedules are among the most complex problems confronting colleges, schools, and educational institutions. In this regard, an attempt is made to assign a collection of specific services, such as classrooms, instructors, and courses, to a set of teaching hours under specific conditions. Universities and schools have different requirements. The allocation of units to classes and professors to units is one of the most significant disparities between universities and colleges. In most colleges, the professor assigned to a course is not specified ahead of time [1, 2].

In other words, a professor can teach multiple courses, while each course is listed in professor schools. In addition,

in colleges, the class assigned to each subject is unknown in advance, while in schools, this is not the case [3].

To put it another way, there are lots of constraints to providing appropriate planning for educational classes. One of them is the availability of professors. Not all instructors are available at all times and offer their free time. Other things like the number of classes and courses are provided. The number of classes, courses, the room's availability, or the class's place must be considered for planning [1].

The following constraint is the students' preferences and conditions. For example, top students finish about four semesters of their studies. In other words, they must pass about 18 courses per semester on average. It is obvious that, in the same semester, all the courses should be planned

differently from one another so that the student can complete the course without any overlapping plans. All these courses are registered as a group. For a specific unit, the number of students in a group becomes very large, and it is impossible to plan them for a specific period in a class. Therefore, holding a parallel class is likely to happen. Suppose 100 students pass a specific course in the same semester, and on the contrary, the capacity of a class is a maximum of 30 students, so four parallel classes will be opened for that course.

It should be noted that the program of that parallel class is not necessarily the same. It depends on the instructor's availability. In addition, there are facilities for a specific course where some classes are merged into one class. Suppose a teacher can teach a number of courses. This makes the timetable more difficult. On the contrary, the timetable of the university program includes several sections. Timetable constraints in the first stage are 8 hours of classes per day and five days per week. Secondly, there are two types of lecturers, full-time and part-time lecturers.

In general, a maximum of 16 theory units per week is planned for a part-time lecturer, while a maximum of 12 units is planned for a full-time lecturer. There is no particular limit for the room where the classroom is held. However, for some specific courses, there is also a workshop and laboratory modes, which have different schedules and make the schedule more difficult [2, 4].

Researchers all around the world have assessed the optimization of curriculum planning and course scheduling in different universities. Cruz-Rosales et al. [3] created a two-stage model in which they first considered a feasible solution using novel approaches such as local search and taboo search and then soft constraints using the simulated annealing process. By simulated annealing (SA) algorithm and graph coloring, they obtained a solvable solution. Next, the SA algorithm then eliminates soft constraints, as well as rearranges time intervals and moves courses between time intervals [3].

A meta-innovative approach based on the forbidden quest was presented by Aldeeb et al. [5]. Different neighborhood structures are discussed in this study. The problem of scheduling courses at the university's department of statistics with four separate neighborhood structures was solved, and the results were compared. Assi et al. [6] provided a model for scheduling the exams at Napier University by removing unscheduled courses using an evolutionary genetic algorithm and local search.

Muklason et al. [7] proposed an integer programming model for the class schedule that ensured the constraints met the requirements. The objective function of this study is to represent the desires of students and professors. Mokhtari et al. [8] provided a study on the topic of nurses' class planning, which is comparable in terms of modeling to educational planning. With the aid of mixed-integer programming, a two-stage creative model was proposed to improve the program's user-friendliness and high versatility in switching between nurses. Luo et al. [9] developed a model to minimize professors' working days and hours of unemployment between the two groups, using simultaneous

linear programming of integers combined with real numbers as a convergent approximation tool to the neighborhood.

After a complete review of the literature, the main contribution of this research can be summarized as follows:

- (i) Optimizing the university class scheduling problem considering the preference of teachers
- (ii) Optimizing the university class scheduling problem considering the preference of students
- (iii) Considering the deviation of ideal plans in optimizing the studied problem
- (iv) Proposing a metaheuristic algorithm to optimize the proposed mathematical model

In the rest of the study and in Section 2, the methodology, including the proposed mathematical model and the genetic algorithm, will be discussed. In Section 3, the numerical results obtained from solving the model are presented, and in Section 4, the research summary is presented.

## 2. Methodology

*2.1. Proposed Mathematical Model.* In this section, an integer linear programming model is presented. In this regard, the principles of proposing a new mathematical model are inspired by [10–14]. Based on modeling from the subject literature [15–17], the objective function of the model is divided into two parts. The first part is for professors, and the second part is for students. Moreover, since it was required to maximize the number of classes in a specific time period, its negative deviations are minimized too. In the following, variables, parameters, and sets are defined. Next, various constraints of the model are presented, and then, the objective of the function and determination of the cost coefficients are presented.

*2.1.1. Indices and Sets.*  $i$ : index of courses

$j$ : index of courses' code

$l$ : index of professors

$d$ : index of days

$t$ : index of periods

$S$ : index of student groups

$l$ : the set of courses that must be presented

$l_1^v$ : a set of one-unit courses (courses that require half a session per week)

$l_2^v$ : sets of two-unit courses (courses that require one session per week)

$l_3^v$ : three-unit course set (courses that require one and a half sessions per week)

$l_4^v$ : four-unit course set (courses that require two sessions per week)

$l_l^L$ : courses that can be presented by the  $L$  professor

$l_s^S$ : courses in which the  $s$ th group of students is applying for registration

$J_i$ : the codes that can be provided for the  $i$ th course

$L_i$ : professors who can teach my course.  
 $T$ : the set of time intervals,  $T = \{0, 1, 2, \dots\}$   
 $TE$ : set of even time intervals,  $T = \{0, 2, 4, \dots\}$   
 $TO$ : set of odd time intervals,  $T = \{0, 1, 3, 5, \dots\}$   
 $TK$ : set of time intervals of  $k$ th time packets

### 2.1.2. Decision Variables

$X_{ijldt}$ : binary variable takes 1 if the code of the class  $i$  is provided at the hour  $t$  of day  $d$ , and is 0 otherwise

$X_{ijldt s}$ : binary variable takes 1 if the code  $j$  of the class  $i$  is provided by the  $l$ th professor at the hour  $t$  of the day  $d$  and the  $s$ th group of students participate in it, and 0 otherwise

$y_{ij}$ : binary variable will take 1 if the code  $j$  of the three-unit class  $i$  has two sessions in an odd week and one session in an even week, and if it has one session in an odd week and two sessions in an even week, it will take 0

$d_{dks}^{s+}$ ,  $d_{dks}^{s-}$ : positive and negative deviations of the number of classes of the  $s$ th group of students in the  $t$ th time block on the day  $d$  from the desired number of classes

$y_{dks}^s$ : binary variable is equal to 1 if the  $s$ th group of students in the  $k$ th time block has class, and 0 otherwise

$d_{dks}^{l+}$ ,  $d_{dks}^{l-}$ : positive and negative deviations of the number of classes of teachers  $l$  in the  $k$ th time block on the day  $d$  from the desired number of classes

$y_{dks}^l$ : binary variable is equal to 1 if a teacher  $l$  has a class in  $k$ th time block on day  $d$ , and 0 otherwise

$z_{ijl}$ : binary variable is equal to 1 if the teacher  $l$  teaches the  $i$ th course, and 0 otherwise

$w_{ijs}$ : binary variable is equal to 1 if group  $s$  of students enrolls in code  $j$  of course  $i$  and 0 otherwise

### 2.1.3. Parameters

$L_l$ : minimum number of hours that the professor  $l$  must teach (for faculty members)

$U_l$ : maximum number of hours that the  $l$ th professor should teach (for professors who are not faculty members)

$a_i$ : capacity of the  $i$ th class (the maximum number of students who can enroll in the  $i$ th class)

$k_s$ : number of students in the  $s$ th student group

$b_{dt}$ : number of available rooms for class scheduling in the  $t$ th interval from the  $d$ th day

$m_{ld}$ : desired number of classes of the  $l$ th teacher in the  $k$ th time package on the  $d$ th day

$n_{ks}$ : desired number of classes of the  $s$ th student group in the  $k$ th time block on the  $d$ th day

$v_i$ : number of units for the course  $i$

$c_{ijldt}$ : cost factor of the formation of the  $J$ th class of the  $i$ th class by the  $l$ th teacher at hour  $t$  of the day  $d$

$c_{ijldt s}$ : coefficient of the cost of forming the class code  $J$  of the  $i$ th class by the  $l$ th teacher at the hour  $t$  of the day  $d$  and registering the  $s$ th group of students in it.

$a_1$ : importance of cost factor in the objective function

$a_2$ : importance of number of classes (negative deviation) in the objective function

$a_3$ : importance of the number of classes (positive deviation) in the objective function

$a_3$ : importance of students' preferences in the class schedule

$a_4$ : importance of teachers' preferences in the class schedule

### 2.1.4. Hard Constraints

(1) *One-Unit Courses*. Constraint (1) has been presented for one-unit courses, which specifies that one of the half-intervals must take the value of one on different days and at different time intervals:

$$\sum_{d \in D} \sum_{t \in T} \sum_{l \in L} x_{ijldt} = 1 \quad \forall i \in I_1^v, j \in J_i. \quad (1)$$

(2) *Two-Unit Courses*. For two-unit courses, Constraints (2)-(3) are used. Constraint (2) makes one of the even half-intervals equal to one and is equivalent to the fact that the first part of a class is formed in one of the even half-intervals, and Constraint (3) equals the next half-interval. It places one so that the continuation of the class is also formed exactly after the first half of the class:

$$\sum_{d \in D} \sum_{t \in T} \sum_{l \in L} x_{ijldt} = 1 \quad \forall i \in I_2^v, j \in J_i, \quad (2)$$

$$x_{ijldt} = x_{ijldt+1} \quad \forall i \in I_2^v, j \in J_i, d \in D, t \in T_E, l \in L_i. \quad (3)$$

(3) *Three-Unit Courses*. As the courses suggest, for the assignment of these courses, three half-intervals of one must be taken. There are two situations here: either two half intervals of even and one half interval of odd value or two odd half intervals and an even one are taken. Constraints (4)-(5) show that if  $y_{ij}$  takes the value of one, we will have two even half-intervals and one odd half-interval, and vice versa, if it takes the value of zero:

$$\sum_{d \in D} \sum_{t \in T_E} \sum_{l \in L} x_{ijldt} = 1 + y_{ij} \quad \forall i \in I_3^v, j \in J_i, \quad (4)$$

$$\sum_{d \in D} \sum_{t \in T_O} \sum_{l \in L} x_{ijldt} = 2 - y_{ij} \quad \forall i \in I_3^v, j \in J_i. \quad (5)$$

Constraint (6) guarantees that two of the three half intervals that are supposed to have a value of one are consecutive so that it can be considered as a full session:

$$y_{ij} - 1 \leq x_{ijldt} - x_{ijldt+1} \leq y_{ij} \quad \forall l \in L_i, t \in T_E, i \in I_3^v, j \in J_i, d \in D. \quad (6)$$

Constraint (7) guarantees that each course does not have more than two half-periods (one session) per day:

$$\sum_{t \in T} x_{ijldt} \leq 2, \forall d \in D, i \in I_3^v, j \in J_i, l \in L_i. \quad (7)$$

(4) *Four-Unit Method.* There are four series of constraints for four-unit courses. Constraint (8) makes two of the even half-intervals equal to one, and it is equivalent to the fact that the first part of a class is formed in two even half-intervals. Moreover, Constraint (9) of their next half-interval is equal to that (one) so that the continuation of the class is formed exactly after the first half of the class:

$$\sum_{t \in T} t x_{ijldt} = 2, \forall i \in I_3^v, j \in J_i, l \in L_i, d \in D, \quad (8)$$

$$x_{ijldt} = x_{ijldt+1}, \forall i \in I_4^v, l \in L_i, t \in T_E, j \in J_i. \quad (9)$$

Constraint (10) guarantees that each four-unit course does not have more than half a period (one session) per day:

$$\sum_{t \in T} t x_{ijldt} \leq 2, \forall i \in I_4^v, j \in J_i, l \in L_i, d \in D. \quad (10)$$

2.1.5. *Constraint of Professors.* Constraint (11) guarantees the noninterference of the courses related to each professor in each time period:

$$\sum_{i \in I} \sum_{j \in J_i} x_{ijldt} \leq 1, \forall l \in L, d \in D, t \in T. \quad (11)$$

Constraint (12) is related to the minimum and maximum teaching hours of professors, which are the minimum teaching hours for faculty members and the maximum teaching hours for some nonfaculty professors who set a maximum time for teaching:

$$L \leq \sum_{d \in D} \sum_{t \in T_E} \sum_{l \in L} \sum_{i \in I_l} x_{ijldt} \leq U, \forall l \in L. \quad (12)$$

Constraints (13)-(14) guarantee that each code teaches each course by one and only one professor. If the  $l$ th teacher teaches the  $j$ th code of the  $i$ th course, the variables related to that teacher can take values. The first Constraint (13) makes it possible to be scheduled in the timetable only when a professor has taught the course code. Furthermore, Constraint (14) guarantees that each course code is taught by only one professor:

$$\sum_{i \in I_l} \sum_{j \in J_i} \sum_{l \in L} x_{ijldt} \leq v_i z_{ijs}, \forall s \in S, i \in I_s^l, j \in J, \quad (13)$$

$$\sum_{l \in L_i} z_{ijl} = 1, \forall i \in I, j \in J_i. \quad (14)$$

2.1.6. *Constraint of Students.* Constraint (15) guarantees the noninterference of courses related to each group of students in each time period:

$$\sum_{i \in I_l} \sum_{j \in J_i} \sum_{l \in L_i} x_{ijldt} \leq 1, \forall s \in S, d \in D, t \in T. \quad (15)$$

Constraint (16) shows that if a course is offered in a day and hour, the number  $a_i$  of students who apply for that course should register in the code of that course:

$$a_i x_{ijldt} = \sum_{s \in S_i} k_s x_{ijldt s}, \forall i \in I, j \in J_i, d \in D, t \in T. \quad (16)$$

Constraints (17) and (18) guarantee that each student group can register one and only one-course code, and if it is registered, the variables related to that code can be set. In fact, the following constraint shows that if course  $i$  was presented with  $v_i$  units for the students of group  $s$ , it will be  $W_{ijs} = 1$ , and the number  $v_i$  of half the interval will be one.

These constraints guarantee that each group of students can register in one and only one of the course codes they want, and if they register, the variables related to that code can be set. In fact, Constraint (17) shows that if course  $i$  of code  $j$  is presented with  $v_i$  units for students of the group  $s$ , it will be  $W_{ijs} = 1$ , and the number  $v_i$  of half interval will be one. Constraint (18) shows that each group of students must register in one code for the courses they want. Moreover, Constraint (19) shows the number of classes available in each time period:

$$\sum_{i \in I_l} \sum_{j \in J_i} \sum_{l \in L_i} x_{ijldt} \leq v_i w_{ijs}, \forall s \in S, i \in I_s^l, j \in J_i, \quad (17)$$

$$\sum_{j \in J_i} w_{ijs} = 1, \forall s \in S, i \in I_s^l, \quad (18)$$

$$\sum_{i \in I_l} \sum_{j \in J_i} \sum_{l \in L_i} x_{ijldt} \leq b_{dt}, \forall d \in D, t \in T. \quad (19)$$

2.1.7. *Soft Constraints.* The soft constraints include the following. Constraint (20) from the professors' point of view shows the minimum number of classes that professors want to have in each time frame package:

$$\sum_{i \in I_l} \sum_{j \in J_i} \sum_{t \in T_k} x_{ijldt} d_{dkl}^{l-} - d_{dkl}^{l+} = m_{kl} y_{kdl}^l, \forall l \in L, k \in K, d \in D. \quad (20)$$

Constraint (21), which is raised from the perspective of students, shows the minimum number of classes that students have in each time package:

$$\sum_{i \in I_i} \sum_{j \in L_i} \sum_{l \in L_i} \sum_{t \in T_k} x_{ijldt} + d_{dkl}^{s-} - d_{dkl}^{s+} = n_{ks} y_{kds}^s, \forall s \in S, k \in K, d \in D. \tag{21}$$

2.1.8. Objective Function.

$$\begin{aligned} Minz = & a_1 \sum_{i \in I_i} \sum_{j \in L_i} \sum_{l \in L_i} \sum_{d \in D} \sum_{t \in T} c_{ijldt} x_{ijldt} \\ & + a_2 \sum_{s \in S} \sum_{d \in D} \sum_{k \in K} d_{dks}^{s-} + a_3 \sum_{s \in S} \sum_{d \in D} \sum_{k \in K} d_{dks}^{s+} \\ & + a_4 \sum_{l \in L} \sum_{d \in D} \sum_{k \in K} \gamma_{dkl}^l + a_5 \sum_{s \in S} \sum_{d \in D} \sum_{k \in K} \gamma_{dks}^s. \end{aligned} \tag{22}$$

Equation (22) shows the objective function of the mathematical model. In general, the different parts of the objective function are measured with coefficients  $a$ , and their relative importance is determined. The first part is for the cost of forming a class, and the second part is the cost of choosing a class of students. The third and fourth sections try to maximize the number of classes in the time package as much as possible, so its negative deviations are minimized. The fifth section makes the class less time-bound.

2.2. The Structure of the Genetic Algorithm. The genetic algorithm (GA) is a classic metaheuristic algorithm based on the evolutionary ideas of natural selection and genetics [18]. In this way, GA represents the intelligent use of a random search algorithm to solve optimization problems. Although genetic algorithms use random phenomena, it has an evolutionary process to achieve the best possible solution [19]. The main aspects of GA are designed in such a way that they can simulate the processes necessary for evolution in natural systems. Among the most important of these processes are those that follow the laws that were first proposed by Charles Darwin and named ‘‘Survival of the Fittest.’’ The reason for this is that, in nature, the competition between living organisms to obtain scarce resources leads to the dominance of the fittest organisms over the weaker ones [20, 21].

In order to solve mathematical models using genetic algorithms, it simulates the principle of fitness with an encoding scheme. It produces several solutions as a ‘‘generation.’’ Each generation contains a population of strings of characters that are similar to the chromosomes found in our DNA. Each person (solution) will represent a point in the search space and a possible solution. Next, the members of each generation enter into a process similar to the process of evolution of living organisms [22].

Genetic algorithms operate using specific principles, such as the genetic structure and behavior of chromosomes in a population of individuals [23]. These basics are

- (i) Selecting a set of solutions and generating new solutions using crossover and mutation
- (ii) Members who are more successful in each competition produce more offspring than members who do not perform well
- (iii) Genes from good-performing members spread within the population, so good parents sometimes produce offspring that are better than either parent

Therefore, each successive generation will be better suited to live in its surroundings. A summary of the steps of the genetic algorithm is presented in Figure 1.

To use a genetic algorithm, a population of individuals is maintained within a search space. Each of these people is actually a possible solution to the problem. Each member is coded with a finite-length vector of components or variables and with the help of special alphabets. These special alphabets are usually binary alphabets. To implement a genetic algorithm structure, each member is interpreted as a chromosome which is known as solution representation. Therefore, a chromosome (a solution to the problem) consists of several genes (variables). To show the abilities of each member to compete with other members of the population, each possible answer is assigned a fitness score. The member with an optimal fitness score (or, more generally, close to optimal) is our favorite [24, 25]. The main reason for choosing GA for optimizing the proposed mathematical model is its acceptable performance in the previous related research items. The flowchart of the proposed genetic algorithm is illustrated in Figure 2.

2.3. Solution Representation. The encryption method used in this study is permutation encryption. In this technique, chromosomes are coded as integers. Each integer is a unique gene for a specific period, and the sequence of integer genes representing genes in a chromosome determines the order of the sequence of periods to be programmed [25]. For example, a five-gene chromosome represented by 4 3 2 1 5 means that the first period is programmed by the integer 4 for the first time. Next, subsequent courses are followed by others in succession. The length of courses is equal to the total number of courses planned in a course group. If there are four periods for planning, and each of them is planned twice a week, the length of the chromosome is 8.

The selection process is based on a short selection. Chromosomes are classified according to their fitness function value from largest to smallest. Some chromosomes, starting from the smallest fitness strands, are replaced by new ones [26]. A new chromosome is obtained by reversing the position of all bits in an old chromosome. Unlike other selection methods, short selection does not copy the best chromosome for the population. For example, Table 1 contains the classified chromosomes of five genes. If the selection probability is 0.4, it means that the number of old chromosomes to be

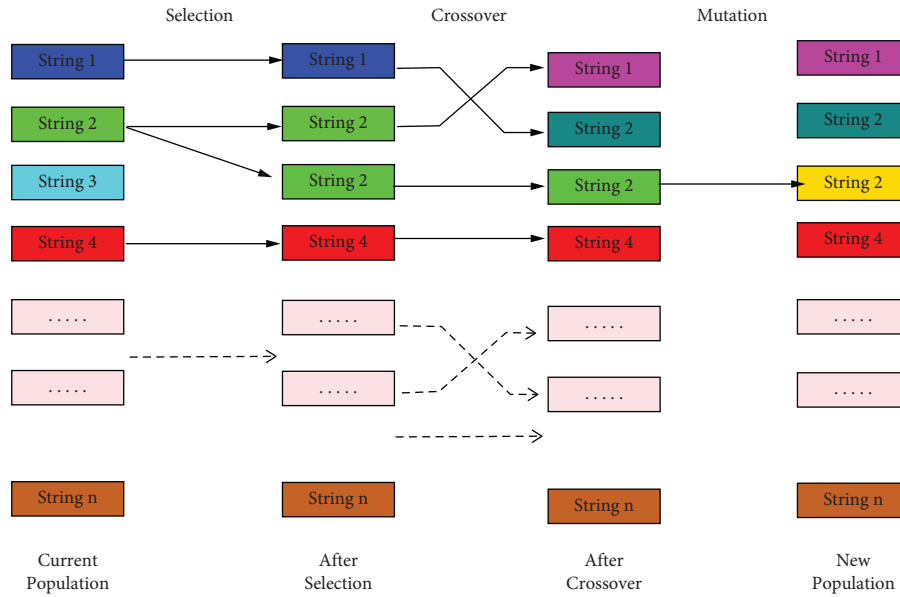


FIGURE 1: The main operators in genetic algorithm.

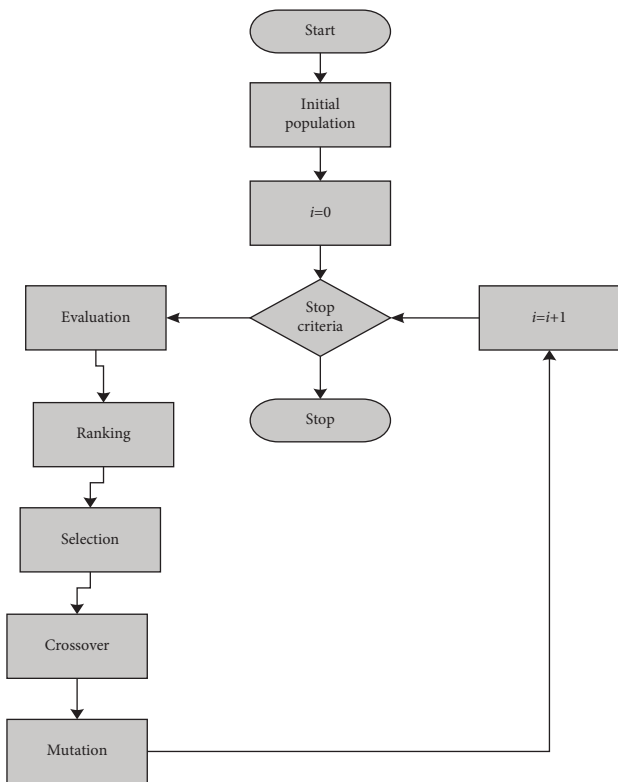


FIGURE 2: The flowchart of the proposed genetic algorithm.

replaced with new ones is  $2 = 0.4 \times 5$ . Chromosomes 1 2 5 3 4 are replaced by 4 3 5 2 1, while chromosomes 5 2 3 4 1 are replaced by 1 4 3 2 5.

**2.4. Crossover Operator.** In this research, a random-based crossover is applied. In the operator, the multiplication of chromosomes is performed based on population size and rate of crossover. The number of crossovers and the number

TABLE 1: Selected chromosomes before crossover.

Chromosomes
1 4 5 2 3 1
3 5 2 1 4 0.95
2 4 5 1 3 0.7
1 2 5 3 4 0.5
5 2 3 4 1 0.3

of mutations must be the same. An example of a crossover is presented in Figure 3.

**2.5. Mutation Operator.** In this research, a random-based mutation is applied. For the mutation operator, two cells are selected randomly; then, the value of these cells are moved with each other. An example of mutation is presented in Figure 4.

**2.6. Timetable Design.** A schedule is actually a schedule that various constraints must accompany. Constraints are almost universally used by people with scheduling problems [16, 18, 20]. Constraints, in turn, are roughly divided into two categories: soft and hard constraints. A hard constraint is a constraint in any work schedule; it does not create any considerations. For example, a lecturer cannot be in two places simultaneously [22, 23]. Soft constraints may be broken, but their violation should be minimized. In addition to constraints, a number of exceptions must be considered when building an automatic scheduling system. For example, each class should be scheduled exactly once, students' classes should not have two instructors at the same time, and reservations should not be made if the instructor is not available.

On the contrary, some classes need special rooms. Some classes must be held consecutively. While soft constraints for

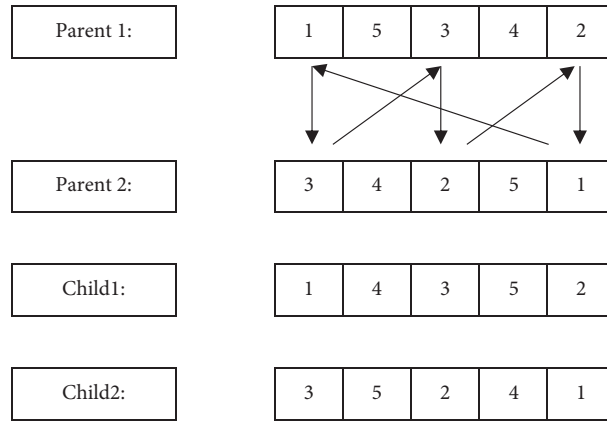


FIGURE 3: An illustrative example of the crossover operator.

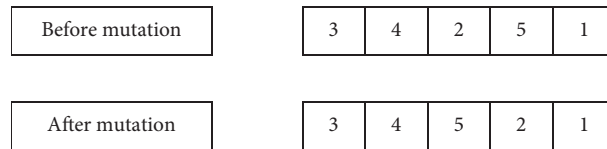


FIGURE 4: An illustrative example of the mutation operator.

some professors prefer hours to be scheduled, most students do not want to have empty courses on their schedule. The distance students have to travel should be kept to a minimum. Classes should be evenly distributed throughout the week, classrooms should be reserved close to their office or home, and classrooms that are much larger than the standard classroom size should be reserved. In addition, there are exceptional limits that have been considered. For example, a part-time instructor can be scheduled for more than six units, while for full-time, it is twelve units.

### 3. Numerical Results

Some tests are conducted to make sure how the system works. There are four groups of courses from year 1 to year 4. In experiments, we want to create a timeline for a particular semester’s data. A certain number of courses and professors are planned.

From Table 2, it can be concluded that  $n(M) = 7$ ,  $n(T) = 3$ ,  $n(L) = 9$ ,  $n(C) = 2$ ,  $n(D) = 5$ , and  $n(H) = 10$ , and all chromosome genes include  $m1, t0, l1, m2, t0, l2, m3, t0, l2, m4, t0, l3, m5, t0, l6, m6, t0, l5, m7, t1, l6, m7, t1$ , and  $l7$ .

This means that there are 13 columns and 100 rows in the target matrix after receiving the basic data, such as what room can be used for a specific teacher in a limited time and day, and also according to the capacity of that room. This system is run with ten chromosomes in a population, and ten generations are set in the experiment. Regardless of the probability of selection, crossover, and mutation, the maximum of the best value of the fitness function, i.e., 1, is obtained. This means that less time is needed to develop the program, and all courses can be planned based on it.

The timetable in which the class session will be held should also be defined. The system can define the available room for it, so after assigning a room, the status of that room is no longer available for that day and time.

**3.1. Findings.** As previously mentioned, the problem is planned and solved in conjunction with real data taken from the Faculty of Electrical Engineering of one of the international universities, which we will discuss below, to discuss the findings of the proposed model. This problem contains five courses with three units, three courses with two units, and one course with one unit, for a total of 39 units (39 half intervals). Table 3 contains a list of courses with descriptions. Furthermore, it has nine professors; each professor can teach a few or no courses. Moreover, Table 4 includes information on professors. These courses are available for twelve different student classes (groups of ten).

Since each class has a capacity of 30 students, each class must enroll three groups of students, as shown in Table 5. In each section, there are between nine and eleven selected units. Every week, we have three working days. It is also considered, given that we have 39 credits and the institute’s objectives of broadcasting activities in the classroom. There are a few details in the model that is not included. There are no minimum or maximum time constraints for professors, for example. All objective coefficients are set to one since we are just searching for a reasonably suitable response and testing the model’s correctness. Two-time packages per day (morning and evening) are considered, with at least three-morning classes and four-evening classes.

The problem’s outputs are provided using the GAMS software. The outputs are divided into three categories after extracting and classifying the variables that take the value:

TABLE 2: The relation between all sets.

Course name		Lecture			Lab		
		A	B	units	A	B	units
ICT	( $m_1$ )	Budi Berlington	( $l_1$ )	3	Monika ( $l_8$ )	Monika ( $l_8$ )	2
Calculus 1	( $m_2$ )	Nababan	( $l_2$ )	2	Finela ( $l_9$ )	Finela ( $l_9$ )	2
Calculus 2	( $m_3$ )	Nababan	( $l_2$ )	2		Andree ( $l_{10}$ )	Andree ( $l_{10}$ )
IPE	( $m_4$ )	Sutrisno	( $l_3$ )	4	Gunawan ( $l_5$ )		
Discrete math	( $m_5$ )	Samuel	( $l_4$ )	4			
Statistics	( $m_6$ )	Gunawan	( $l_5$ )	3			
Reading skills	( $m_7$ )	Univ ( $l_6$ )	Univ ( $l_7$ )	2			2

TABLE 3: The available list of courses and their specifications.

Course code	Course title	Code of professors who can teach	Number of course units	Groups of applicant students	Number of codes available
1	Electric car 3	1 and 6	3	11-9-8-5-4-3	2
2	Airline design and project	1	3	1-5-10	1
3	Special electric machines	2 and 8	3	12-10-9-6-5-1	2
4	Power system analysis	2 and 4	3	2-6-7-8-9-11	2
5	Special topics	4 and 5	3	2-3-4-6-7-12	2
6	Technical reporting	3	3	7-4-2	1
7	Digital systems 3	3	2	1-3-5-8-9-11	2
8	Numerical calculations	7	2	2-3-7-4-10-12	2
9	Ethics	9	1	1-6-8-10-11-12	2

TABLE 4: List of students and their characteristics.

	Elective course codes	Collect unit
1	2-3-7-9	9
2	4-5-6-8	10
3	1-5-7-8	10
4	1-5-6-8	11
5	1-2-3-7	10
6	3-4-5-9	10
7	4-5-6-8	9
8	1-4-7-9	9
9	1-3-4-7	11
10	2-3-8-9	9
11	1-4-7-9	9
12	2-3-8-9	9

TABLE 5: List of available professors and their specifications.

Professors code	Course codes	Available time periods
1	1-2	Day 1 and morning of day 3
2	3-4	Day 3 and morning of day 2
3	4-5	Day 2
4	5	Morning of day 2 and morning of day 3
5	8	Day 1
6	3	Morning of day 1 and morning of day 2
7	9	Evening of day 3



TABLE 6: Schedule class variables.

	0	1	2	3	4	5	6	7	8	9
Day 1			$X_{31812}$		$X_{11114}$	$X_{11115}$	$X_{12116}$	$X_{12117}$	$X_{21118}$	$X_{21119}$
Day 2	$X_{31820}$ $X_{41220}$	$X_{52521}$ $X_{42221}$	$X_{51522}$	$X_{32223}$ $X_{51523}$	$X_{72324}$	$X_{72325}$	$X_{72326}$	$X_{71327}$	$X_{61328}$	$X_{61329}$
Day 3	$X_{11130}$ $X_{32230}$	$X_{51531}$ $X_{32231}$	$X_{21133}$ $X_{52533}$	$X_{12132}$ $X_{52532}$	$X_{42234}$	$X_{422355}$	$X_{41236}$	$X_{41237}$		

TABLE 7: Schedule of the institution.

	8–10 AM	10–12 AM	2–4 PM	4–6 PM	6–8 PM
Day 1	Numerical calculations code 1	Special electric machines code 1	Electric car 3 code 1	Electric car 3 code 2	Airline design and project code 1
Day 2	Operations research code 1 ( <i>f</i> )	Electric machines for code 2 ( <i>g</i> ) special topics for code 1	Electricity generation code 2	Electricity generation code 1	Technical reporting code 1
Day 3	Special topics code 2 ( <i>g</i> )	“Electric machines 3 code 2 ( <i>F</i> )”	Code analysis of power system code 2	Code analysis power system 1	—

TABLE 8: Assignment of courses to professors.

Row	Course	Course code	Course professor code	Professor
1	Electric machine 3	Code 1	1	A
2	Electric machine 3	Code 2	1	A
3	Airline design and project	Code 1	1	A
4	Special electric machines	Code 1	8	B
5	Special electric machines	Code 2	2	C
6	Power system analysis	Code 1	2	C
7	Power system analysis	Code 2	2	C
8	Special topics	Code 1	5	D
9	Special topics	Code 2	5	D
10	Technical reporting	Code 1	3	E
11	Electricity generation	Code 1	3	E
12	Electricity generation	Code 2	3	E
13	Numerical topics	Code 1	8	F
14	Numerical topics	Code 2	8	F
15	Ethics	Code 1	9	G
16	Ethics	Code 2	9	G

the first category is the institution’s results, the second category is the professors’ results, and the third category is the students’ results. Table 6 will be used to display the following results.

After converting the final solution to the schedule, we reach the following schedule. It should be noted that there is a break between 3 and 4 (two hours for lunch from 12 to 14). The results are presented in Tables 7 and 8.

### 4. Conclusion

The scheduling problem is one of the issues that has been the focus of many researchers in recent years. This problem is used in different fields in the real world. One of the practical aspects in which the scheduling problem is used is the scheduling problem of university courses, which is a complex decision-making issue at the beginning of each academic semester for university teaching groups. Educational groups in universities have always been in pursuit of providing courses and having a suitable timetable schedule

according to the needs and educational laws. This issue has received more attention in recent years, and today, it has been raised as a decision-making issue. One of the most challenging tasks for university administrators and planners is scheduling university courses. The presence of multiple and overlapping constraints in planning is the primary cause of these difficulties. On the one hand, students and professors each have their own set of constraints. As an influential factor, university rules can impose constraints on planning in this region.

The proposed genetic algorithm and heuristic search are able to solve the timetable scheduling problem in academic institutions. The main finding of this research is that using the proposed approximate solution method can provide some solutions that satisfy all constraints, including the students’ and professors’ preferences. In this regard, in the optimal solution, planning includes allocating the courses to the available time periods which are achieved in a way that student satisfaction is maximized. Although some entities, such as class location, are not included in the target matrix,

the system is able to determine which room to use for a specific cell in a timeline. If it is placed in the target matrix, it is likely that instead of adding the number of columns, a three-dimensional target matrix will be created.

On the other hand, there are limitations in this study that will be solved as a problem in the future. First, each parallel class of a subject group shown in a target matrix must be scheduled for each period defined in that timetable. Secondly, a course is only possible if it has at least the number of course units in the target matrix. If that period is divided into two parts, the name of that period must be different, and it can be assumed as two periods. Third, the system's main goal is to maximize the number of successful units in planning. Otherwise, it should be designed based on that definition.

On the contrary, using the proposed mathematical modeling tools is suggested as an effective method for scheduling university classes. This model helps in the optimization of mathematical relationships in order to achieve the best possible solution to a variety of problems. The results of this study show that the use of optimization tools can create a straightforward program with all possible constraints.

## Data Availability

The data used to support the findings of the study can be obtained from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] E. K. Burke, D. G. Elliman, and R. Weare, "A university timetabling system based on graph colouring and constraint manipulation," *Journal of Research on Computing in Education*, vol. 27, no. 1, pp. 1–18, 1994.
- [2] M. Kaur and S. Saini, "A review of metaheuristic techniques for solving university course timetabling problem," *Advances in Information Communication Technology and Computing*, pp. 19–25, 2021.
- [3] M. H. Cruz-Rosales, M. A. Cruz-Chávez, F. Alonso-Pecina et al., "Metaheuristic with cooperative processes for the university course timetabling problem," *Applied Sciences*, vol. 12, no. 2, p. 542, 2022.
- [4] M. Abolghasemian, A. P. Chobar, M. AliBakhshi, A. Fakhr, and S. Moradi, "Delay scheduling based on discrete-event simulation for construction projects," *Iranian Journal of Operations Research*, vol. 12, no. 1, pp. 49–63, 2021, p.
- [5] B. A. Aldeeb, M. A. Al-Betar, A. O. Abdelmajeed et al., "A comprehensive review of uncapacitated university examination timetabling problem," *International Journal of Applied Engineering Research*, vol. 14, no. 24, pp. 4524–4547, 2019.
- [6] M. Assi, B. Halawi, and R. A. Haraty, "Genetic algorithm analysis using the graph coloring method for solving the university timetable problem," *Procedia Computer Science*, vol. 126, pp. 899–906, 2018.
- [7] A. Muklason, R. G. Irianti, and A. Marom, "Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm," *Procedia Computer Science*, vol. 161, pp. 656–664, 2019.
- [8] M. Mokhtari, M. Vaziri Sarashk, M. Asadpour, N. Saeidi, and O. Boyer, "Developing a model for the university course timetabling problem: a case study," *Complexity*, vol. 2021, Article ID 9940866, 12 pages, 2021.
- [9] X. Luo, Y. Sun, X. Liu, Y. Gao, H. Sun, and Y. Liu, "Course timetable optimization for a university teaching building considering the building energy efficiency and time-varying thermal perception of students," *Building and Environment*, Article ID 109175, 2022.
- [10] E. B. Tirkolae, A. Goli, P. Ghasemi, and F. Goodarzian, "Designing a sustainable closed-loop supply chain network of face masks during the COVID-19 pandemic: pareto-based algorithms," *Journal of Cleaner Production*, vol. 333, Article ID 130056, 2022.
- [11] E. Asgharizadeh, M. Kadivar, M. Noroozi, V. Mottaghi, H. Mohammadi, and A. P. Chobar, "The intelligent traffic management system for emergency medical service station location and allocation of ambulances," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2340856, 12 pages, 2022.
- [12] S. Jahangiri, A. Pourghader Chobar, P. Ghasemi, M. Abolghasemian, and V. Mottaghi, "Simulation-based optimization: analysis of the emergency department resources under COVID-19 conditions," *International Journal of Industrial and Systems Engineering*, vol. 1, no. 1, pp. 1–10, 2021.
- [13] A. Babaeinesami, H. Tohidi, P. Ghasemi, F. Goodarzian, and E. B. Tirkolae, "A closed-loop supply chain configuration considering environmental impacts: a self-adaptive NSGA-II algorithm," *Applied Intelligence*, vol. 52, pp. 13478–13496, 2022.
- [14] N. Abolfazli, M. Eshghali, and S. F. Ghomi, "Pricing and coordination strategy for green supply chain under two production modes," in *Proceedings of the 2022 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 13–18, IEEE, Virginia, USA, 2022, April.
- [15] H. Algethami and W. Laesanklang, "A mathematical model for course timetabling problem with faculty-course assignment constraints," *IEEE Access*, vol. 9, Article ID 111666, 2021.
- [16] C. Akkan, A. Gülcü, and Z. Kuş, "Bi-criteria simulated annealing for the curriculum-based course timetabling problem with robustness approximation," *Journal of Scheduling*, vol. 4, pp. 477–501, 2022.
- [17] A. Pourghader chobar, M. Sabk Ara, S. Moradi Pirbalouti, M. Khadem, and S. Bahrami, "A multi-objective location-routing problem model for multi-device relief logistics under uncertainty using a meta-heuristic algorithm," *Journal of Applied Research on Industrial Engineering*, vol. 9, no. 3, pp. 354–373, 2022.
- [18] A. Pourghader Chobar, M. A. Adibi, and A. Kazemi, "A novel multi-objective model for hub location problem considering dynamic demand and environmental issues," *Journal of industrial engineering and management studies*, vol. 8, no. 1, pp. 1–31, 2021.
- [19] A. Goli, H. K. Zare, R. Moghaddam, and A. Sadeghieh, "A Comprehensive Model of Demand Prediction Based on Hybrid Artificial Intelligence and Metaheuristic Algorithms: A Case Study in Dairy Industry," *Journal of Industrial and Systems Engineering*, vol. 11, 2018.
- [20] A. Goli, H. Khademi-Zare, R. Tavakkoli-Moghaddam, A. Sadeghieh, M. Sasanian, and R. Malekalipour Kordestanizadeh, "An integrated approach based on artificial intelligence and novel meta-heuristic algorithms to predict demand

- for dairy products: a case study,” *Network: Computation in Neural Systems*, pp. 1–35, 2020.
- [21] P. Ghasemi, H. Hemmaty, A. Pourghader Chobar, M. R. Heidari, and M. Keramati, “A multi-objective and multi-level model for location-routing problem in the supply chain based on the customer’s time window,” *Journal of Applied Research on Industrial Engineering*, 2022.
- [22] A. Goli and H. Mohammadi, “Developing a sustainable operational management system using hybrid Shapley value and Multimoora method: case study petrochemical supply chain,” *Environment, Development and Sustainability*, vol. 24, pp. 10540–10569, 2022.
- [23] A. Goli, H. Khademi Zare, R. Tavakkoli-Moghaddam, and A. Sadeghieh, “Hybrid artificial intelligence and robust optimization for a multi-objective product portfolio problem Case study: the dairy products industry,” *Computers & Industrial Engineering*, vol. 137, Article ID 106090, 2019.
- [24] A. P. Chobar, M. A. Adibi, and A. Kazemi, “Multi-objective hub-spoke network design of perishable tourism products using combination machine learning and meta-heuristic algorithms,” *Environment, Development and Sustainability*, pp. 1–28, 2022.
- [25] W. Erben and J. Keppler, “A genetic algorithm solving a weekly course-timetabling problem,” in *Proceedings of the International Conference on the Practice And Theory Of Automated Timetabling*, pp. 198–211, Springer, Gent, Belgium, 1995, August.
- [26] J. Zhao, “Sumoylation regulates diverse biological processes,” *Cellular and Molecular Life Sciences*, vol. 64, no. 23, pp. 3017–3033, 2007.