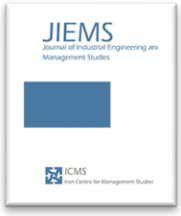
Vol. 9, No. 8, 2022, Pages 95-108

JIEMS
Journal of Industrial Engineering and Management Studies

journal homepage: www.jiems.icms.ac.ir



Improved satisfaction of university faculty by utilizing the Bat metaheuristic algorithm (Case study of the faculty of humanities, Islamic Azad University, Parand Branch)

Alireza Ariyazand[1], Hamed Soleimani[*1,2], Farhad Etebari[1], Esmaeil Mehdizadeh[1]

[1]Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.
[2]School of Mathematics and Statistics, University of Melbourne, Melbourne, Parkville, VIC 3010, Australia.

Received: Dec 2021-22/ Revised: Feb 2022-22/ Accepted: Mar 2022-19

Abstract

Scheduling is a vital part of daily life that has been the focus of attention since the 1950s. Knowledge of scheduling is a very important and applicable category in industrial engineering and planning of human life. In the field of education, scheduling, and timetabling for best results in classroom teaching is one of the most challenging issues in university programming. As each university has its own rules, policies, resources, and restrictions a unique model of scheduling and timetabling cannot implement. This can cause more complexity and challenging point which needs to be considered scientifically. This study presents a sound scientific model of timetabling and classroom scheduling to improve faculties' desirability based on days, times, and contents preferences. A sample in Parand branch of Islamic Azad university chooses using the Bat metaheuristic algorithm. By considering the limitations, some unchangeable constraints regarding the specific rules and minimal linear delimitation of the soft constraints of the model, using the appropriate meta-heuristic algorithm to reduce the model run time to a minimum. The results show that the algorithm achieves better results in many test data compared to other algorithms due to meeting many limitations in the problem coding structure. The Bat algorithm is compared with four other algorithms while comparing the results of solving the proposed mathematical model with five metaheuristic algorithms to evaluate the performance. In this research, a multi-objective model is presented to maximize the desirability of professors and to solve the model using Bat, Cuckoo Search, Artificial bee colony, firefly, and Genetic algorithms. In this research 40 different runs of each algorithm were compared, and conclusions were drawn. Modeling has been solved with GAMS and MATLAB software and using the bat meta-heuristic algorithm. It is concluded that in this model, the bat algorithm is the most appropriate algorithm with the shortest time, which has caused the satisfaction of the professors of the educational departments of this academy.

Keywords: university course timetabling; multi-objective model; bat meta-heuristic algorithm.

Paper Type: Original Research

1. Introduction

Timetabling problem, in short, means allocating activities to limited resources (human, instrumental, spatial, and temporal (available to optimally perform tasks in a way that prevents duplication) (Veenstra and Vis 2016).Types of timetabling problem are classified in different ways, one of which includes: 1- Course timetabling problem (Post et al., 2014) Examinations timetables problem (Di Gaspero et al., 2007)Project scheduling problem (Agarwal, Colak and Erenguc, 2011) Employee service scheduling problem(Barrera, Velasco & Amaya, 2012) Transportation and public transport scheduling problem (Shafia et al, 2012) scheduling of call centers and Post 7 - Flight scheduling(Pita, Barnhart and Antunes, 2012) schedule of Police Centers 9 schedule of Medical Services(Jafari and Salmasi, 2015)1510 - schedule of Sports Competitions(Lewis and Thompson, 2011).

Although university timetabling has been studied a lot, but its clinical application has been less of an area of attention.

One of the issues in which timetabling problem is very important is the issue of timetabling problem of university courses, which is a complex decision-making issue and carries some limitations as well. Since the most important factors to achieve a successful curriculum planning and programming in each university are its faculties' satisfaction of their schedule and timetabling based on their preferences without any overlapping or, program interference. To prevent the time interference of the two programs with each other, certain restrictions should be met, and the satisfaction of the employer and the client should be obtained (Ahmed, Özcan, & et al. 2012).

. in this study we aimed to provide a suitable mathematical model according to the available resources and limitations, to increase the satisfaction of faculties of Parand School of Humanities as a sample of this case study, based on three criteria (day, time, and subjects). As in multi-objective optimization problems, the goals may conflict with

*Corresponding Author: hd_soleimani@yahoo.com

DOI: 10.22116/iiems.2022.316347.1468

each other and the optimization of one goal will weaken the other goals. Therefore, a logical approach to multi-objective problems can be considered as obtaining a set of Pareto optimal solutions instead of a single solution (Akkan, and018 Gulcu 2). Since the weekly planning courses of each unit is based on various factors such as: university policies and rules, number of units in each curriculum (Maine and elective), number of available classes in each time, days of availability of faculties and their subject expertise, capacity and prerequisites of each course, student groups and variety of fields and educational levels. in some cases, the need to change the curriculum and multi-objective model to ensure the desirability of professors and their preferences, encouraged us to use mathematical modeling based on research objectives and limitations and to use meta-heuristic algorithms. five Meta-heuristic algorithm and the Bat Algorithm, which has advantages and shorter implementation times than other run algorithms. Considering its importance in generalizing to other faculties and university units, the Faculty of Humanities of the Islamic Azad University, Parand unit, where Mr. Ariyazand taught as one of the researchers from 2010 to 2015, was selected for case study. MATLAB and GAMS software were used for solving the model and analyzing the results.

2. Literature review

This group behavior-based metacognitive optimization algorithm is widely used to solve multi-objective optimization problems and is currently used as one of the most powerful methods for solving many complex optimization problems. This nature-inspired algorithm was first introduced in 2010 by Yang, X.S35 to simulate the collective behavior of bats, the only winged mammals that use sound reflection to hunt their prey. The bat algorithm is based on the use of sound reflection in a natural environment as an optimization algorithm based on collective intelligence and intelligent behavior of bats. The bats find the exact path and location of their prey by sending sound waves and receiving its reflection, and when sound waves return to the transmitter (bat).
This bird can use the desired waves to draw an acoustic image of obstacles in front of its surroundings and even in complete darkness to identify its surroundings well and recognize moving objects around the environment such as insects and fixed objects in the environment like trees by using this system. This algorithm is based on the echo detection feature of micro bats. Bats are generally classified into two types: large-bats and micro-bats. The micro-bats use echo reflection (sound reflection) to fly at night and hunt, which is a perceptual system in which ultrasonic waves are generated to echo. The bat's brain and nervous system can compare the transmitted waves and the reflected waves, create an image of the surrounding space with its details and identify its prey in complete darkness with the help of this ability.
The wave intensity produced by the bat is 130 decibels, which uses a frequency of 15 to 20 KHz to hunt prey. The range of human hearing is from 20 Hz to 20 KHz. In order to be able to identify the data obtained, bats must separate the sound produced by them from its echo. Micro-bats can do this in two ways: 1- Echo detection (sound reflection) with cycles with short time intervals: This group of bats can distinguish their sent sound from the reflected sound with the help of timing. 2- Echo detection (sound reflection) with cycles with long time interval: This group of bats, while producing long-term continuous sound, separates the pulses and echoes by changing the frequency, and this group of bats are able to change each production frequency based on the flight speed of the pulse so that the received echo is still in the appropriate hearing range.
The ideal rules of the bat algorithm include the following three rules:
A) All bats use echo detection feature Sound reflection (can estimate distance and difference between food) Prey and fixed obstacles ahead.
B) Bats randomly at Vi speed in location Xi with constant frequency fmin with variable wavelength λ and loudness sound A0 are in search of prey and can fully automatically adjust the wavelength of their emitted pulses and match their pulse emission rate that is $r \in (0,1)$ based on the proximity of their prey.
C)Although the volume can be changed in different ways, it is assumed that this volume changes from a large (positive value) A0 to a minimum constant value of Amin, and generally the frequency f is in the distance [fmin – fmax] which corresponds to the wavelength spectrum as [λmin-λmax].
The bat algorithm is introduced to solve continuous optimization problems, but if it is necessary to search in discrete space, we use the correct type of bat algorithm and use a decimal fraction transfer function to convert the particle position to zero or one and the new position of the bats is updated as an integer.
Despite extensive studies and research in the field of timetabling problem, according to experts in this field, there are still many shortcomings that these shortcomings are addressed in two ways: 1- Developing models closer to real problems 2- Improving timetabling problem solving methods. Considering the comparison of the present study with the researches done in the past, it is observed that the present study, in addition to trying to improve the timetabling problem of courses of Parand Faculty of Humanities, tries to provide maximum satisfaction to the professors of this faculty. This will address one of the shortcomings in this area. While the bat algorithm is new and efficient for solving the problem of timetabling problem of university courses, considering the shortcomings

in solving the problem of timetabling problem of university courses under conditions of limited resources, it can be concluded that there is a need to develop models close to real problems. The present research tries to eliminate this shortcoming by creating a suitable model for the mentioned problem. Here are some of the researches that have been done so far on the issue of lesson timetabling problem under conditions of limited resources. In general, during more than half a century, various types of university course timetabling problems have been presented, which are different in terms of the university in question and the rules and methods of solving many models, such as mathematical, computer, graph, etc. It has been presented in the past that the purpose of all of them is to provide a program that by entering new information, the optimal program is tailored to the goals, needs and resources of the Research University. This ideal optimal program is a big topic on the basis of operations research. This is being used, in train travel times, city traffic control, or at schools or universities. Both time and place play a role in this matter. Efforts should be made to make best use of all classes and resources. This situation is slightly different at school versus university. At university, allocating classes to students and faculty is different compared to school. At university, until schedule day, no one knows what service are available and also there are many more units at university compared to school The mathematical models, especially linear models, have a special place in this field, and in 1998, Wood and Whitaker. Presented a Nonlinear model for timetabling problem of high school classes. In 2001, Dimopoulou and Miliotis12 introduced an integer programming model for assigning courses to time periods and classes with the aim of increasing the desirability of assigning training programs to time intervals. University course timetabling, is one of the most important topics of resource allocation.
In 2004, Daskalaki10 et al. proposed a method for calculating the Zero-One and one-time timetabling problems for university course timetabling problem, in which the desirability of assigning courses to different time periods varies. Daskalaki has set aside time to allocate courses for certain time periods and seeks to minimize these costs. The issues of timetabling problem of university courses from the aspect of modeling research in operations of artificial intelligence are very attractive for researchers, which has led to a lot of research in this field in recent years (Bolaji et al. 2014)9. In general, the timetable of university courses includes the timetable of courses, students, professors and classes in a fixed number of time periods, taking into account the limitations of the university Branch (Basir, Ismail, & Norwawi 2013). Marriott & Stuckey mentioned the ability to plan constraints to execute applications using more than 3,000 constraints as one of the superior features of constraint planning. In his research, Lü & Hao18 presented a comparative forbidden search algorithm to solve the lesson timetabling problem in 2010, which resulted in the optimization of this algorithm by implementing a specific data set and comparing it with other algorithms (Lü, Z., & Hao, J.-K. 2010). Daskalaki is one of the people who has made a lot of efforts in modeling classroom integers. He first solved the problems of timetabling problem of high school classes and then presented the integer programming models of timetabling problem of university courses and implemented them as a case study. Ranjbar and Rostami (2012) developed a model for university timetabling in Ferdowsi university. They used a linear absolute number model for their timetabling[25].

(Badoni, Gupta & Mishra 2014) proposed a new linking algorithm combining genetic and local search algorithms to define events based on grouping students with algorithm performance based on values of different factors (population, combination and crossover) to solve timetabling problem problems. Turning the problem into three smaller problems allocating, distributing, and timetabling problem of courses so that the solution space decreases and speeds it up (Rangel-Valdez et al.2014). Development of Link Bee Community Optimization Algorithm Solving University Lesson Timetabling Problems Presented by Alzaqebeh& et al (2015). Phillips et. al. described a university timetabling model in Auckland university. They used this model to solve the problem of assigning classes in larger scale university settings using an absolute number model (Phillips et. al. 2015). University professors in Brazil (Pereira and Gomes Costa, 2016). Vermonten et. al. looked at using an absolute number model to decrease the movement of students from class to class while marinating faculty satisfaction and reducing the number of days that faculty would need to work. This was achieved by allocating lessons to specific time periods and available classes (Vermonten et. al). In 2017, Song et al, developed a timetabling problem algorithm that helps college administrators and planners save energy on campus. In 2017, Goh and colleagues combined two local algorithms to solve the post-enrollment timetable. Fonseca[13] et al., 2017, proposed new sections and preprocessing techniques and modified the original formula for the curriculum problem. Since the purpose of the timetable problem is to assign a set of lessons to retransmission and class, Bagger et al., 20186, divided these issues into two parts: timetabling problem and class assignment. They used the port algorithm to generate gaps that linked class timetabling problem and allocation, Nagata also used local search in 2018 to solve the post-enrollment timetable problem. In 2020, Landir[16] et al. developed an integer programming model in which high school parallel collaboration and faculty preferences were assigned by allocating lessons to time periods and available classes. Teachers were also assigned to periodic lessons to maximize teacher's satisfaction and productivity. Shahmoradi et. al. explained timetabling for university lessons using AHP method by means if OPL program on Cplex. They considered soft

limitations of the model and assumed minimal distraction from other models (Shamoradi et. al. 2018). Tavakoli has looked at two different mathematical models for university timetabling. One model was a general model while the other considered some limitations and specifics. Such specifics were: knowledge superiority of some professors compared to others, time assigned to each professor, faculty consultation to students, quality of teaching. They found out that the model that considered specifics, was superior (Tavakoli et. al. 2018).

3. Problem statement

In many cases, the issue of timetabling problem of university courses has been studied in theory. However, in the form of practical planning of courses, it is observed that not only paying attention to the limitations and resources to solve such problems is enough, but also the preferences of professors and universities should be considered in the timetable. Considering that the planning of weekly courses of university Branches is based on various factors such as country laws, university policies, large volume of courses, many resources and restrictions such as available classes, time limitation for classes, access to professors and their free hours, variety of basic courses , Main and optional, facilities and facilities required for each course, capacity and prerequisites of each course, student groups and variety of fields and educational levels and in some cases the need to change the curriculum and multi-objective model to ensure the desirability of professors and their preferences, encouraged us to use mathematical modeling based on research objectives and limitations and to use meta-heuristic algorithms to solve models.
The purpose of the problem is to present the model of timetabling problem of university courses in a faculty in such a way that the desirability of professors is maximized during the classes and the courses offered within the existing constraints.
While observing the number of course credit per class, 16 hours of educational hours (45 hours per educational hour) during a semester is considered, which includes 6 working days per week and each day includes six times a day Two hours each time from 8am to 8: 30pm.
In setting up the model, we are faced with the limitations of available resources such as professors, classrooms, and timetabling, including working days and one-day time periods, and it is not possible to deviate from the following basic rules:
In general, the Mathematical model limitations included in the model are as follows:
1- Classes with specific facilities and capacity are considered and the number of students in each classroom cannot exceed a final ceiling based on the capacity of the class.
2- The time of presence of professors at the university, the courses they are able to teach and the teacher's preference for teaching the course are specified by default.
3- All time periods are the same and the days of classes are clear and fixed.
4- Course groups and lessons related to each course group are clear and fixed.
5- Non-interference of classes, lessons, lecturers and class groups.
6- Professors are different in terms of different criteria (academic rank, service history and teaching of the desired course, type of employment, etc.) and have different weights, and the number of students in each department is specific and fixed.
In addition, soft restrictions should be observed as much as possible, although in some cases, professors or departments may be violated for some courses, which are usually accompanied by fines, such as maximizing the desirability of professors, offering different courses, minimizing disruption in the program. Minimize relocation and change of classes and teachers and the like.
Planning constraints (regulations limitations in timetabling design) include:
A) All sessions related to a lesson and group must be assigned to a teacher.
B) The minimum and maximum number of sessions allowed for each professor in the current semester must be observed according to the laws of the Ministry of Science and the policies of the university and the relevant faculty, which can be different for different professors.
C) All sessions related to a lesson and group must necessarily be assigned to one teacher.

4- Mathematical model of the problem:

removes unacceptable combinations from the problem by defining each activity based on them. An example of an unacceptable combination is a professor or group of students that are not defined for a specific class or lesson. The final result of this reduces the problem size and the number of variables dramatically.
In this section, model is formulated through multi-objective binary integer programming, in which each activity includes a professor, a day, a course, a class group of students in a specified period of time, and the required facilities are assigned to each course. In case of interference, planning is not feasible and due to interference, it will not be possible to hold classes. For example, if more than one professor or a group of students is assigned to a classroom in a period of time, it will interfere and it will not be possible to hold a course in that classroom.

The sets parameters and decision variables are presented:

Sets:

r={1,…,R}	The set of classrooms
d ={1,…,D}	The set of classes' day per week
t ={1,…,T}	The set of periods Of Time (time intervals) of classes per day
c ={1,…,C}	The set of courses
p ={1,…P}	The set of professors
g ={1,…,G}	The set of students' groups
f ={1,…,F}	The set of Facilities required for each course

Parameters:

v_{pc}:	Desirability of professor p to choose a course c
v_{pd}:	Desirability of professor p to choose the day d
a_p:	The importance of professor p for university
n_c:	The Number of units of course c
l_p:	The Minimum number of units assigned to professor P
u_p:	The Maximum number of units assigned to professor p
h^p_{max}:	The Maximum number of units can be assigned to professor p in one day
h^p_{min}:	The Minimum number of units can be assigned to professor p in one day
i_{gr}:	The Maximum capacity of students' group g in classroom r
j_{pd} $\begin{cases}1\\0\end{cases}$	If the professor p has the class on the day d Otherwise
b_{pc} $\begin{cases}1\\0\end{cases}$	If the professor p presents the course c Otherwise
q_{pd} $\begin{cases}1\\0\end{cases}$	If the professor p is present on the day d Otherwise
e_{cd} $\begin{cases}1\\0\end{cases}$	If course c is available to be presented on day d Otherwise
φ_{rf} $\begin{cases}1\\0\end{cases}$	If the classroom r has facilities f Otherwise
z_{cf} $\begin{cases}1\\0\end{cases}$	If course c requires facilities f Otherwise

Variable:

x_{cpgtdr}: A binary variable that if professor p presents course c for students' group g in period of time t of day d in classroom r, value will be 1, otherwise will be 0.

The objective functions:

$$\text{Maxf}_1 = \sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} a_p . v_{pc} . x_{cpgtdr} \quad (1)$$

$$\text{Maxf}_2 = \sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} a_p . v_{pd} . x_{cpgtdr} \quad (2)$$

1-Objective function (1): Maximizes professors' desirability in selecting the courses to be taught Objective function.

2- Objective function (2): Maximize professors' desirability in choosing the working days Teaching

Constraints:

$$\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{t=1}^{T}\sum_{d=1}^{D} x_{cpgtdr} \le i_{gr} \; ; \; \forall g, r \quad (3)$$

Constraint (3) ensures that the number of students in the students' group assigned to each class must be less than or equal to the capacity of that class.

$$\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{c}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr} \le t \; ; \; \forall t \quad (4)$$

Constraint (4) ensures that all courses must be timetabled according to the number of daily time's periods (ultimately equal to the daily time periods).

$$\sum_{g=1}^{G}\sum_{p=1}^{P} x_{cpgtdr} \le 1 \; ; \; \forall c, d, t, r \quad (5)$$

Constraint (5) confirms that only one lesson is offered in a classroom at a time of a day.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{r=1}^{R} x_{cpgtdr} \le 1 \; ; \; \forall p, t, d \quad (6)$$

Constraint (6) ensures that a professor must deliver only one course at a period of Time in a day.

$$\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{r=1}^{R} x_{cpgtdr} \le 1 \; ; \; \forall g, t, d \quad (7)$$

Constraint (7) ensures that each group of students could only be in one classroom at a specific time in a day.

$$\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr} \le b_{pc} \; ; \; \forall p, c \tag{8}$$

Constraint (8) ensures that a course will not be assigned to a professor that does not wish to teach that course.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr}.n_c \ge l_p \; ; \; \forall p \tag{9}$$

Constraint (9 according to university rules, units assigned to each professor should be greater than or equal to the minimum units assigned to each professor(L_p).

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr}.n_c \le u_p \; ; \; \forall p \tag{10}$$

Constraint (10) according to university rules, units assigned to each professor should be less than or equal to the maximum units assigned to each professor (U_p).

$$\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D} x_{cpgtdr}.z_{cf} \le \varphi_{rf} \; ; \; \forall c, r, f \tag{11}$$

Constraint (11) ensures that a course requiring special facilities should be held in a class classroom that e desired facilities.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{r=1}^{R} x_{cpgtdr}.n_c \le j_{pd}.q_{pd}.h_{max}^{p} \; ; \; \forall p, d \tag{12}$$

Constraint (12) indicates that if the professor p attends the university on day d, if he / she has a course, he / she will be assigned maximum of H_{max}^{p} units.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{r=1}^{R} x_{cpgtdr}.n_c \ge j_{pd}.q_{pd}.h_{min}^{p} \; ; \; \forall p, d \tag{13}$$

Constraint (13) indicates that if the professor p attends the university on day d, if he / she has a course, he / she will be assigned minimum of H_{min}^{p}units.

$$\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{r=1}^{R} x_{cpgtdr} \le e_{cd}.r.t \; ; \; \forall c, d \tag{14}$$

Constraint (14) indicates that course c could only be offered on day d if the group manager allows it on that day.

5. Solution approach

Since the studied problem is known as NP-Hard type, a novel meta-heuristic algorithm is suggested. As the proposed mathematical model is multi-objective study problem, multi-objective optimization methods should be employed. Thus, the Bat algorithm method is developed.

For this reason, in this research, the problem with bat algorithms, cuckoo search algorithm, Artificial bee colony algorithm (MOABC), firefly algorithm (MOFA) and genetic algorithm (MONSGA-II) solved and after comparison, according to the results obtained the bat algorithm, MOBAT better results and had less running time than other algorithms. The bat algorithm is based on tracking characteristics and search method of bat hunting.

Bats can track and prey on their prey in complete darkness by emitting sound and receiving it, and the following three rules are used to develop this algorithm.

●All bats use sound reflection to detect distances and know the difference between food and obstacles in front of them.

●Bats fly of stochastically v_t speed in x_t place at a constant frequency f_{min} and different wavelengths λ and loudness *A*0 for prey hunting. They can automatically propagate waves and adjust their transmitted pulse rates (r∈[1,0]) according to the proximity of their prey.

●Given that the volume may vary in many different ways, we assume that the volume varies from R0 (maximum value) up to R_{min} to minimum value (variable).

Based on the above rules, the location x_i^t and velocity v_i^t for each virtual bat *i* are calculated in repetition t and frequency f_i based on the following formulas:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{15}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*) \tag{16}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{17}$$

It should be noted that β∈[0,1] is a random vector with a uniform distribution and x^* is the best current location selected in each iteration after comparison with the virtual bat position.

The frequency f is considered with $f_{min} = 0$ and $f_{max} = 100$ in the local search in each iteration one of the answers is selected as the best answer and the new position of each bat is updated locally and in a random step with the following formula:

$$x_{new} = x_{old} + \varepsilon a_0^t \quad (18)$$

Where $\varepsilon\epsilon$ [−1,1] is a random number and a_i^t is the average loudness of bats is in repetition t and the loudness of a_i and the pulse rate sent r per repetition are updated with the following formulas:

$$a_i^{t+1} = \alpha a_i^t \quad (19)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \quad (20)$$

The values α and γ are fixed values that 0 <α <1 and γ > 0 that

And we have: $a_i^{t+1} \to 0$ and $r_i^{t+1} \to r_i^0$ when $t \to \infty$

In addition, in order to enhance the search in the problem-solving space. Crossover and mutation users are also being used. Also since the mathematical model is multi-objective Four different other algorithms have been used to solve the problem and achieve the best Pareto surface.

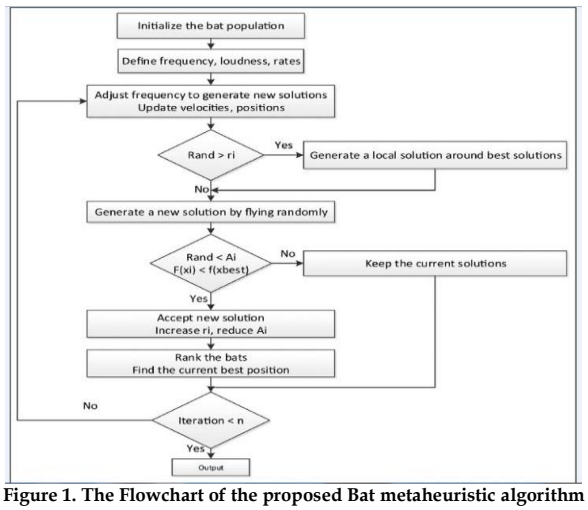


Figure 1. The Flowchart of the proposed Bat metaheuristic algorithm

6. Algorithm comparison and results

This article, while paying attention to the general educational environment and existing laws and regulations in Iran, examines the multi-objective mathematical model of university course planning with the aim of maximizing the satisfaction and desirability of Parand Azad University's humanities faculty members. It is attractive, but there are limitations in the parameters, such as the topics of a course that may vary during different semesters (depending on the time of the semester) or the level of satisfaction of the professors, which is actually a relative factor.

It should be noted that the augmented ε-constraint method is coded by the GAMS software package. Table1 indicates the results obtained by both augmented ε-constraint method and Bat algorithm for five small scale instances. All data, for small scale instances, are generated randomly.

Table 1. The results of augmented ε-constraint and BAT Algorithm for small scale instances

Instance	augmented ε-constraint		BAT Algorithm	
	OF 1	OF 2	OF 1	OF 2
1	11.5106	5.9987	11.4991	6.0003
2	9.8901	6.0031	9.7608	5.9563
3	11.3029	6.4201	10.9013	6.3408
4	10.6349	5.8031	9.6937	5.5136
5	8.7091	5.0384	8.9743	4.9604

From Table 1, which is solved using GAMS software, two important results are obtained. First, the mathematical model developed for the problem has sufficient validity. Second, the chosen meta-heuristic algorithm (Bat) has the necessary efficiency for issues related to increasing global desirability. Now, the proposed algorithm is solved by examining the case study, using MATLAB software, and the results of solving the model with the Bat meta-heuristic algorithm will be compared with four other meta-heuristic algorithms to evaluate the effectiveness of this algorithm compared to other algorithms. Required software in one semester in the Faculty of Humanities of Islamic Azad University, Parand Branch are provided:

In the studied semester, 2132 students are studying in this faculty in the form of 5 educational groups (3 discipline in associate degree, 2 discipline bachelor's degree, 5 Bachelor's degree and 3 discipline in master's degree).

Classes of this faculty are held six days a week (from Saturday to Thursday) with 41 classes. The time of each session in this research is 150 minutes, 135 minutes for the class, and 15 minutes to change classes. One hour a day is dedicated to worship (lunchtime & prayer), Thus, 4 periods are considered. The lessons presented in the semester included 489 lessons per week which are offered by 146 professors including: laws 42 lecturers, accounting 51

lecturers, English language21 lecturers, physical education 17 lecturers and public management 15 lecturers The classes are divided into 726 class groups will be taught in the current semester.
Class facilities including: white boards, maps and graphs, audio recordings, video, computer, video projectors, workshop and gym, these are divided into 24 different forms of facilities for holding educational classes at the Faculty of Humanities, Islamic Azad University, Parand branch, The minimum numbers of units allowed (required) is 16 hours per week for an instructor, 15 hours for an assistant professor, 14 hours for an associate professor and 12 hours for a professor. However, at the discretion of the university, they can participate in other non-educational activities for up to half of these hours, and faculty members can teach up to 8 hours in the mentioned unit, and invitee lecturers can teach up to 18 hours in a university branch. Have teaching hours per week.
To compare the algorithms from each of the algorithms of bat, cuckoo, bee, genetics and firefly forty different run with a npop 10 and maxiter 100 are solved with MATLAB software version 2018 and after solving the algorithms, issues such as Time required to run the algorithm run, Number Of Solutions(Pareto surface points)(NO),the mean ideal distance (MID), the maximum spread of diversity(MD) and spacing have been compared and each multi-objective algorithm with its best objective value and the minimum time required for converge of each algorithm was tested. The Bat algorithm, considering the coefficients A = 0.9 and r = 0.5 among the solution results of the five algorithms in total, had less solution time and better results than the other algorithms. This algorithm is able to find the most Pareto surface between 6 to 8 points (with the shortest running time among the five algorithms (between 74.1737 to 127.0692) with an average time of 118.974 seconds). While it also offers the best maximum distribution range and ideal distance, and the implementation of this algorithm on the planning of the Faculty of Humanities, Islamic Azad University, Parand branch (as a case study) increased the satisfaction of the professors of this university branch. The results are shown in Tables 2-6.

Table 2. The results of MO-Cuckoo Search (Npop10, Maxiter100)

Row	No	MID	MD	Spacing	Min Objs	Max Objs	Running time (s)
1	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	363.783461
2	5	4.0774	4.614	1.1122	5.4921,5.5734	10.6914,6.63733	400.8565
3	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	79.568108
4	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	379.645034
5	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	103.847153
6	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	91.761645
7	3	2.7873	1.6347	0.49447	9.7217,4.8321	10.7213,6.56012	328.675304
8	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	376.872618
9	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	369.742382
10	4	3.2965	1.5524	0.66155	9.4901,6.3407	10.8263,6.58821	391.487532
11	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	378.784195
12	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	76.424295
13	5	4.4695	2.0975	0.69355	8.9306,6.1848	10.1304,7.34639	384.097853
14	3	3.4505	1.4627	0.47734	9.8328,5.8146	10.1869,7.19569	349.861074
15	4	3.3386	2.5232	0.89747	8.2823,5.8396	10.7437,6.30606	389.540619
16	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	369.817604
17	3	3.1526	1.4521	0.49995	9.3597,5.2993	9.803,6.6217	381.019625
18	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	439.730608
19	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	354.851006
20	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	379.907516
21	3	4.4857	1.7376	0.49928	8.3075,6.3384	10.4623,6.67999	349.590827
22	3	3.4505	1.4627	0.47734	9.8328,5.8146	10.1869,7.19569	372.874516
23	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	393.700625
24	3	3.7186	2.1084	0.57048	8.3705,6.0552	10.8797,6.49637	426.716495
25	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	369.008306
26	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	381.746105
27	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	358.872615
28	3	4.381	2.2562	0.81377	8.9004,6.2806	11.3158,7.45174	364.392583
29	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	372.493578
30	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	369.902864
31	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	371.703165
32	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	381.740935
33	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	376.876551
34	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	69.871425
35	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	423.782506
36	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	375.374643
37	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	374.640291
38	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	391.0829745
39	4	3.3386	2.5232	0.89747	8.2823,5.8396	10.7437,6.30606	376.375914
40	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	348.302416

According to Table 2, The results of Cuckoo Search algorithm, the average time of forty Runs is 340.983 seconds. The No in this algorithm is between three to five points (with an average of 3.63). The average of MD in this algorithm is 1.996. The average of MID is 3.896, that is better than other algorithms. The average of spacing in this algorithm is 0.599 that is better than NSGA-II, MOFA, and MOBAT algorithms.

Table 3. The results of MOABC

Row	No	MID	MD	Spacing	Min Objs	Max Objs	Running time (s)
1	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	246.8326
2	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	317.3508
3	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	289.6528
4	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	341.7492
5	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	243.7891
6	2	3.1845	1.0636	0	8.5055,5.0231	7.3763,5.0919	302.6487
7	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	246.8645
8	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	236.9035
9	2	3.3889	1.0455	0	8.4808,4.9227	8.3409,6.0068	251.3849
10	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	239.8963
11	4	5.4175	2.0127	0.66577	6.1937,4.97	8.2104,5.3709	328.648
12	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	293.0384
13	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	311.5831
14	2	12.108	1.7153	0	6.0457,5.5746	8.9866,5.6608	246.9031
15	3	12.304	2.03	0.49886	6.5483,4.953	8.8454,6.2509	243.4152
16	2	4.7807	1.1914	0	7.728,4.6578	7.3844,6.0349	281.5482
17	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	232.1736
18	2	3.8733	1.0321	0	7.6152,5.5497	8.6732,5.6735	331.6457
19	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	274.6281
20	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	314.7106
21	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	243.5279
22	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	294.3614
23	2	3.3889	1.0455	0	8.4808,4.9227	8.3409,6.0068	297.1641
24	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	318.4102
25	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	263.796
26	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	313.0461
27	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	271.0485
28	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	233.6597
29	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	217.0465
30	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	306.8954
31	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	258.6743
32	2	3.2261	1.3764	0	7.995,5.6	9.8892,5.662	242.7538
33	2	3.3889	1.0455	0	8.3409,4.9227	8.4808,6.0068	238.4281
34	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	251.4095
35	2	5.565	1.2487	0	6.5583,5.3901	8.1138,5.4967	281.3724
36	3	3.5288	1.83	0.51297	7.4733,4.5978	9.0416,5.8339	264.4271
37	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	231.7639
38	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	271.7643
39	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	246.8761
40	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	259.315

According to Table 3, The results of Artificial bee colony algorithm, the average time of forty Runs is 272.028 seconds, that is better than MOCS and NSGA-II algorithms. The No in this algorithm is between two to five points (with an average of 3.75) that is better than MOCS algorithm. The average of MD in this algorithm is 2.068 that is better than MOCS algorithm. The average of MID is 4.969 that is better than NSGA-II and MOFA algorithms. The average of spacing in this algorithm is 0.445993 that is better than other algorithms.

Table 4. The results of NSGA-II (Npop10, Maxiter100)

Row	No	MID	MD	Spacing	Min Objs	Max Objs	Running time (s)
1	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	178.8173
2	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	299.5238
3	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	228.02845
4	5	3.9181	2.8683	0.70012	8.0683,4.7999	10.5764,6.6487	116.9831
5	7	4.6879	3.7751	0.92378	7.3946,5.9659	11.3059,6.5856	138.9473
6	6	4.0077	3.5917	0.7201	8.4707,4.6429	11.9317,7.20338	189.0438
7	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	201.7542
8	6	3.3512	3.8905	0.85921	7.8537,5.6416	11.2213,6.47552	159.5432
9	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	176.0841
10	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	148.6306
11	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	263.0528
12	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	214.3905
13	5	3.9181	2.8683	0.70012	8.0683,4.7999	10.5764,6.6487	137.8793
14	7	4.6879	3.7751	0.92378	7.3946,5.9659	11.3059,6.5856	199.6931
15	6	4.0077	3.5917	0.7201	8.4707,4.6429	11.9317,7.20338	186.0841
16	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	205.7654
17	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	393.8657
18	5	3.391	2.1665	0.86032	9.9962,5.3578	10.7906,7.39146	531.2945
19	7	16.0888	3.8328	0.85107	5.9392,5.6395	10.1077,7.41005	498.7253
20	6	3.2371	3.7244	0.78799	9.0366,5.311	12.424,7.1722	506.6341
21	7	2.935	2.8852	0.81116	9.8125,5.359	11.6404,6.99803	403.6436
22	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	486.7231
23	5	4.1015	2.5362	0.74556	8.422,6.269	11.2025,6.77113	523.6573
24	6	4.4439	3.8968	0.77636	7.5828,5.3655	12.3226,6.72705	459.1923
25	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	432.1958

26	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	194.2984
27	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	214.2842
28	4	3.8595	2.2698	0.61121	9.0961,5.748	10.9224,7.27629	193.2781
29	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	201.5963
30	5	3.5989	1.9087	0.69614	8.9175,5.9949	10.1635,6.60344	187.1594
31	8	4.248	3.8987	0.94461	8.1551,6.0233	10.9129,7.04679	216.3928
32	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	506.1983
33	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	413.7824
34	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	526.8257
35	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	482.5297
36	4	7.4373	2.6886	0.64809	7.0484,6.0356	10.0677,6.53241	278.1065
37	5	5.8652	2.5387	0.74632	7.5319,5.7651	10.3406,6.65241	214.9583
38	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	318.5873
39	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	213.8678
40	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	461.3745

According to Table 4, The results of NSGA-II algorithm, the average time of forty Runs is 297.585 seconds that is better than MOCS algorithm. The No in this algorithm is between five to eight points (with an average of 6.125) that is better than MOCS and MOABC algorithms. The average of MD in this algorithm is 3.222 that is better than MOCS and MOABC algorithms. The average of MID is 6.437. The average of spacing in this algorithm is 0.801 that is better than MOFA algorithm.

Table 5. The results of MOFA (Npop10, Maxiter100)

Row	No	MID	MD	Spacing	Min Objs	Max Objs	Running time (s)
1	5	3.9181	2.8683	0.70012	8.0683,4.7999	10.5764,6.6487	243.9682
2	4	7.4373	2.6886	0.64809	7.0484,6.0356	10.0677,6.53241	246.8152
3	5	5.8652	2.5387	0.74632	7.5319,5.7651	10.3406,6.65241	273.5682
4	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	236.8675
5	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	158.3748
6	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	178.3753
7	6	5.1547	3.3873	0.79894	7.8203,5.817	11.3681,6.97566	191.4874
8	6	4.5308	3.8622	0.77555	8.2265,5.561	12.2317,7.26957	289.0465
9	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	203.4872
10	6	2.2777	2.8134	0.88923	7.8203,5.817	11.3681,6.97566	195.4174
11	7	6.4592	4.564	0.8562	5.6175,5.3944	10.4446,7.18135	156.6273
12	4	7.4373	2.6886	0.64809	7.0484,6.0356	10.0677,6.53241	146.7423
13	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	164.4036
14	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	172.859
15	4	7.4373	2.6886	0.64809	7.0484,6.0356	10.0677,6.53241	142.6296
16	5	5.8652	2.5387	0.74632	7.5319,5.7651	10.3406,6.65241	145.6072
17	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	181.6187
18	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	167.0934
19	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	137.6743
20	6	5.1547	3.3873	0.79894	7.8203,5.817	11.3681,6.97566	142.7502
21	6	4.5308	3.8622	0.77555	8.2265,5.561	12.2317,7.26957	170.0216
22	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	148.8046
23	6	2.2777	2.8134	0.88923	7.8203,5.817	11.3681,6.97566	184.9862
24	7	6.4592	4.564	0.8562	5.6175,5.3944	10.4446,7.18135	149.6471
25	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	182.6485
26	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	268.1749
27	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	161.9485
28	6	3.2761	2.9969	0.77808	8.1045,4.7514	10.3654,6.3309	186.4569
29	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	149.5392
30	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	174.0785
31	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	168.9434
32	6	3.2761	2.9969	0.77808	8.1045,4.7514	10.3654,6.3309	146.4683
33	7	4.6879	3.7751	0.92378	7.3946,5.9659	11.3059,6.5856	139.7149
34	6	4.0077	3.5917	0.7201	8.4707,4.6429	11.9317,7.20338	147.8442
35	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	154.8764
36	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	126.8439
37	5	4.1015	2.5362	0.74556	8.422,6.269	11.2025,6.77113	141.9847
38	6	4.4439	3.8968	0.77636	7.5828,5.3655	12.3226,6.72705	158.9213
39	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	179.8665
40	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	173.8362

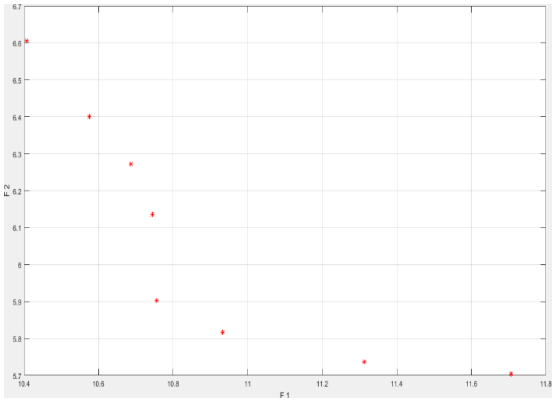
According to Table 5, The results of Firefly algorithm, the average time of forty Runs is 177.276 seconds, that is better than MOCS, MOABC and NSGA-II algorithms. The No in this algorithm is between four to eight points (with an average of 6.15) that is better than MOCS, MOABC, and NSGA-II algorithms. The average of MD in this algorithm is 3.225 that is better than MOCS, MOABC, and NSGA-II algorithms. The average of MID is 6.421 that is better than NSGA-II algorithm. The average of spacing in this algorithm is 0.803.

Table 6. The results of MOBAT (Npop10, Maxiter100)

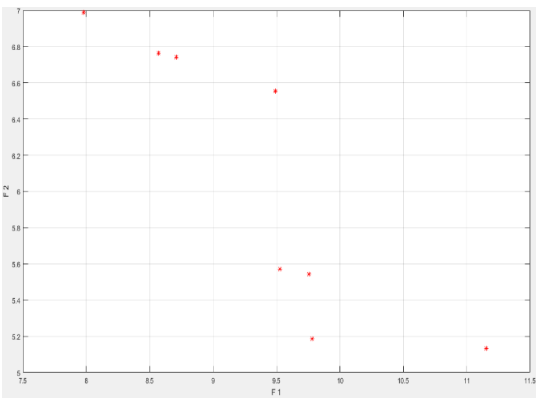
Row	No	MID	MD	Spacing	Min Objs	Max Objs	Running time (s)
1	8	2.6082	2.4168	0.78436	10.4074,5.7045	11.7067,6.60454	122.0737
2	8	4.4942	3.6871	0.83563	7.9794,5.1325	11.153,6.98907	120.1722
3	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	122.5063
4	6	2.6953	2.4909	0.78537	10.1909,5.26984	11.6201,6.79966	124.3175
5	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	118.1486
6	7	5.1424	3.7966	0.80294	7.9295,5.1109	9.9115,7.0229	124.475
7	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	119.2948
8	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	117.6195
9	7	4.844	1.9245	0.90738	8.0468,5.8209	9.2141,6.4878	123.5522
10	8	4.4942	3.6871	0.83563	7.9794,5.1325	11.153,6.98907	121.1945
11	6	2.6953	2.4909	0.78537	10.1909,5.26984	11.6201,6.79966	124.1785
12	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	119.5752
13	8	3.433	2.541	0.80308	8.2204,5.3453	9.3869,6.168	112.4115
14	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	117.0987
15	7	4.3564	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	117.2135
16	7	5.1424	3.7966	0.80294	7.9295,5.1109	9.9115,7.0229	127.0692
17	8	3.433	2.541	0.80308	8.2204,5.3453	9.3869,6.168	115.2699
18	7	5.1424	3.7966	0.80294	7.9295,5.1109	9.9115,7.0229	74.1737
19	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	119.5752
20	6	2.6953	2.4909	0.78537	10.1909,5.26984	11.6201,6.79966	125.3264
21	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	118.1708
22	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	120.282
23	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	118.4048
24	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	117.2726
25	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	118.6684
26	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	122.8511
27	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	116.9844
28	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	117.9403
29	7	5.2079	3.2339	0.7974	7.9191,5.3821	10.7043,6.95074	118.9419
30	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	119.8093
31	6	2.6953	2.4909	0.78537	10.1909,5.26984	11.6201,6.79966	123.2292
32	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	124.0004
33	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	117.5131
34	7	5.1424	3.7966	0.80294	7.9295,5.1109	9.9115,7.0229	124.8175
35	8	4.4942	3.6871	0.83563	7.9794,5.1325	11.153,6.98907	120.6327
36	6	2.9439	1.0357	0.51964	8.6897,5.213	8.9963,5.7364	115.3263
37	8	3.433	2.541	0.80308	8.2204,5.3453	9.3869,6.168	118.6181
38	6	3.9546	3.024	0.76773	7.671,4.8087	9.7713,6.3617	117.6303
39	7	5.1424	3.7966	0.80294	7.9295,5.1109	9.9115,7.0229	125.2062
40	8	2.9783	4.219	0.83763	9.0525,5.3886	12.6561,7.11805	117.4162

According to Table 6, The results of Bat algorithm, the average time of forty Runs is 118.974 seconds, that is better than other run algorithms (less than Cuckoo Search, Artificial bee colony, NSGA-IIand Firefly algorithms). The No in this algorithm is between six to eight points (with an average of 7.075) that is better (more) than the Pareto surface of other run algorithms. The average of MD in this algorithm is 3.945, that is better than MOABC, NSGA-II and MOFA algorithms. The average of MID is 3.264 that is better than other run algorithms. The average of spacing in this algorithm is 0.799, that is better (less) than MOFA and NSGA-II algorithms.

a)



b)



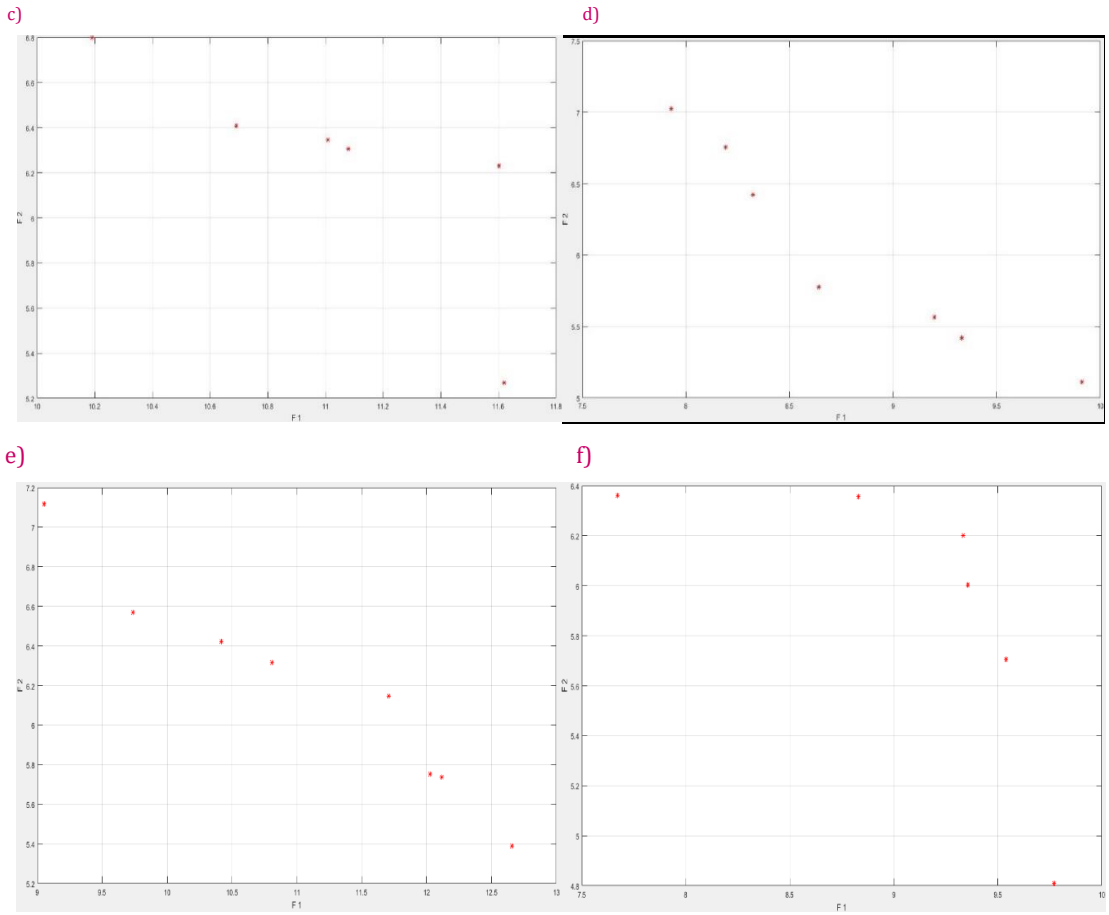


Figure 2: indicates six different runs of the BAT Algorithm on Pareto front

Figure 2 shows some examples of Pareto surface diagrams of answers of MATLAB software runs using bat meta-heuristic algorithm. Due to the similarity of the output diagrams of the Pareto surface, diagram (a) related to the runs 1 is (with 8 points) and diagram (b) related to the runs 2,10 and 35 (with 8 points) and diagram(c) related to runs4,11,20and31(with 6 points) and diagram(d) for runs6,16,18,34 and 39 (with 7 points) and diagram(e) for runs5,7,21,24,26 ,30 32, and 40 (with 8 points) and the graph(f) related to the runs 3,14,22,25, 28,33and 38 (with 6 points).

The desired algorithm found these points effectively and in a short time and as observed in the review and comparison of the run 5 tables of the algorithms, the bat algorithm provides the most points of the Pareto surface with a minimum of 6points and a maximum of 8 points and the best maximum spread range of points of compared to other algorithms which has the best and most effective output of the Pareto surface points among the five run algorithms.

7. Conclusion

University timetabling is a complicated multi objective topic that lacks an aim function and is not something that could be generalized to other universities. Limitations, regulations and other differences between universities or countries will make drawing any conclusion, difficult. Besides, personal preferences of faculty, students and university staff is also different and will make things more complicated. In order to solve these problems, previous year timetabling or trial and error is used which has a lot of flaws and is a less than ideal method to solve the problem. These methods do not take into consideration, some limitations and thus, are not designed to achieve any desirability or satisfaction. In our research, prior to designing a model, we looked at faculty satisfaction and allocated a number to their satisfaction. Then a mathematical model was designed based on soft and hard limitations using GAMS software as a trial run. Then 5 metaheuristic algorithms were used by MATLAB software to design the main model. Forty runs of each algorithm were performed and results were compared between the algorithms. We concluded that the Bat metaheuristic algorithm is superior to all other algorithms in time required to run the model, number of ideal answers, distribution and overall data gathering. This might not stand true in other models or other universities due to inter-university differences and other regulations.

7. Future research

1. Optimization of multi-objective mathematical model timetabling problem of university courses in order to reduce planning time and university preferences using new meta- heuristic algorithms.
2. Optimization of multi-objective mathematical model timetabling problem of university courses with the aim of increasing the desirability of students, departments and university preferences in different academic levels in a university unit.
3. Optimization of multi-objective mathematical model of university course timetabling problem with the aim of increasing students' academic efficiency and their learning efficiency and the preferences of professors in departments in an academic unit.
4. Presenting a mathematical model to minimize the waste of time holding educational classes and compressing the curriculum of the educational departments of university units.
5. Optimizing the mathematical model of timetabling problem of university courses in order to minimize the use of educational space and classrooms for units such as scientific universities, applications that face space constraints.
6. Presenting a two-stage mathematical model, including first assigning a course to each semester and then planning to increase the desirability of university students or to reduce the planning time and increase the desirability of the university unit.
7. Presenting a mathematical model to determine the days of separate presentation of university courses for different educational levels so that the collage days of students of each educational level become different from other levels.
8. Presenting a mathematical model to determine the days of specialized and general courses related to each group and educational level based on the suggestions of students, especially in the graduate course.
9. Timetabling problem of university courses using multi-stage models to facilitate model solving.
10. Presenting a mathematical model of university course timetabling problem for the integration of common and specific course classes of university units that have not reached the quorum due to the lack of welcome from students.

References

Agarwal, A., Colak, S., & Erenguc, S. (2011). A neurogenetic approach for the resource-constrained project scheduling problem. Computers & operations research, 38(1), 44-50.

Ahmed, L.N., Özcan, E. and Kheiri, A. (2012). Solving high school timetabling problems worldwide using selection hyper-heuristics. Expert Systems with Applications. 42(13), 5463-5471.

Akkan, C. and Gulcu, A. (2018). A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem. Computers & Operations Research. 90(1), 22-32.

Alzaqebeh, M., & Abdullah, S. (2015). Hybrid bee colony optimization for examination timetabling problems. Computers & Operations Research, 54, 142-154.

Badoni, R. P., Gupta, D. K., & Mishra, P. (2014). A new hybrid algorithm for university course timetabling problem using events based on groupings of students. Computers & Industrial Engineering, 78, 12-25.

Bagger, N. C. F., Sørensen, M., & Stidsen, T. R. (2018). Benders' decomposition for curriculum-based course timetabling. Computers & Operations Research, 91, 178-189.

Barrera, D., Velasco, N., & Amaya, C. A. (2012). A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation. Computers & Industrial Engineering, 63 (4), 802-812.

Basir, N., Ismail, W., & Norwawi, N. M. (2013). A simulated annealing for Tahmidi course timetabling. Procedia Technology, 11(1), 437-445.

Bolaji, A.L., Khader, A.T., Al-Betar, M.A. and Awadallah, M.A. (2014). University Course Timetabling using Hybridized Artificial Bee Colony with Hill Climbing Optimizer, Journal of Computational Science, 5(5), 809-818.

Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. European Journal of Operational Research, 153(1), 117-135.

Di Gaspero, L., McCollum, B., & Schaerf, A. (2007). The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's University, Belfast, United Kingdom.

Dimopoulou, M., & Miliotis, P. (2001). Implementation of a university course and examination timetabling system. European Journal of Operational Research, 130(1), 202-213.

Fonseca, George HG, et al. Integer programming techniques for educational timetabling. European Journal of Operational Research 262.1 (2017): 28-39.

Goh, S. L., Kendall, G., & Sabar, N. R. (2017). Improved local search approaches to solve the post enrolment course timetabling problem. European Journal of Operational Research, 261(1), 17-29.

Jafari H, Salmasi N. Maximizing the nurses' preferences in nurse scheduling problem: mathematical modeling and a meta-heuristic algorithm. Journal of Industrial Engineering International. 2015 Sep 1;11(3):439-58.

Landir et. al. 2020 Saviniec, L., Santos, M. O., Costa, A. M., & dos Santos, L. M. (2020). Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems. European Journal of Operational Research, 280(3), 1064-1081.

Lewis R, Thompson J. On the application of graph colouring techniques in round-robin sports scheduling. Computers & Operations Research. 2011 Jan 31;38(1):190-204.

Lü, Z., & Hao, J.-K. (2010). "Adaptive tabu search for course timetabling". European Journal of Operational Research, 200(1), 235-244.

Nagata, Y. (2018). Random partial neighborhood search for the post-enrollment course timetabling problem. Computers & Operations Research, 90, 84-96.

Pereira, V., & Gomes Costa, H. (2016). Linear integer model for the course timetabling problem of a faculty in Rio de Janeiro. Advances in Operations Research, 2016.

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

Pita, J. P., Barnhart, C., & Antunes, A. P. (2013). Integrated flight scheduling and fleet assignment under airport congestion. Transportation Science, 47(4), 477-492.

Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., ... & Schaerf, A. (2014). XHSTT: an XML archive for high school timetabling problems in different countries. Annals of Operations Research, 218 (1), 295-301.

Rangel-Valdez, N., Jasso-Luna, J. O., Rodriguez-Chavez, M. H., & Bujano-Guzman, G. (2014)., Practical relaxation of a special case of the Curriculum-Based Course Timetabling problem, Progress in Artificial Intelligence, 2(4), 237-248.

Ranjbar, M., Rostami, S., "Fortnightly Course Scheduling Problem: A Case Study", 2nd International Industrial Engineering Conference, The 5th Int. Conference of The Iranian Society of Operations Research, 2012.

Reisman, A., Kumar, A., & Motwani, J. (1997). Flowshop scheduling/sequencing research: A statistical review of the literature, 1952-1994. IEEE transactions on Engineering Management, 44(3), 316-329.

Shafia, M. A., Aghaee, M. P., Sadjadi, S. J., & Jamili, A. (2012). Robust Train Timetabling problem: Mathematical model and Branch and bound algorithm. Intelligent Transportation Systems, IEEE Transactions on, 13 (1), 307-317.

Shahmoradi Hadi, Saeedeh's ketabi, Ismailian Majid (2018), timetabling problem of university courses using constraint planning, production management and operations.

Song, K., Kim, S., Park, M., & Lee, H. S. (2017). Energy efficiency-based course timetabling for university buildings. Energy, 139, 394-405.

Stafford Jr, E. F., & Tseng, F. T. (2002). Two models for a family of flowshop sequencing problems. European Journal of Operational Research, 142(2), 282-293.

Tavakoli,M. M.,SHIROUYEHZAD H.,HOSSEINZADEH LOTFI F.,NAJAFI E(20018), Proposing A New Mathematical Model For Planning Of University Course Timetabling Based On Quality Of Lesson Presentation, JOURNAL OF OPERATIONAL RESEARCH AND ITS APPLICATIONS,15(3),45-66.

Veenstra, M. and Vis, I.F.A. (2016). School timetabling problem under disturbances. Computers & Industrial Engineering, 95(1) 175-186.

Vermuyten, H., Lemmens, S., Marques, I., & Beliën, J. (2016). Developing compact course timetables with optimized student flows.European Journal of Operational Research, 251(2), 651-661.

Wood, J., & Whitaker, D. (1998). Student centred school timetabling. Journal of the Operational Research Society, 49(11), 1146-1152.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.

This article can be cited: Ariyazand, A., Soleimani, H., Etebari, F., Mehdizadeh,E.,(2022). Improved satisfaction of university faculty by utilizing the Bat metaheuristic algo-rithm (Case study of the Faculty of Humanities, Islamic Azad University, Parand Branch). Journal of Industrial Engineering and Management Studies, Vol.9, No. 8, pp. 95-108.