Vol. 10, No. 2, 2024, Pages 131-148

JIEMS
Journal of Industrial Engineering and Management Studies

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

Optimization of a multi-objective university course timetabling problem with a hybrid WOA&NSGA-II (Case study: IAU, Robat Karim 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, VIC3010, Australia.

Received: Sep 2023-09/ Revised: Jan 2024-19/ Accepted: Jan 2024-22

Abstract
Scheduling and timetabling for university system have been a source of attention and an important challenge for the people in charge of administrations. The regulations and infrastructures are very diverse between universities, making it impossible to come up with a universal model for all. We, in this research, focused on coming up with an algorithm to help with timetabling of class courses for Islamic Azad university of Robat Karim. Our goal was to define an algorithm that could improve teacher satisfaction, and overall efficiency of the university timetabling. Instead, we managed to come up with an efficient algorithm. This research considers different factors such as teacher satisfaction, knowledge and skillset, categorizes students based on undergraduate versus post graduate degree, their research background, their scores and finally student satisfaction as well. This multi-objective mathematic model accounts for all the rules, regulations, and limitations of the university setting while following challenging confinements that guarantee the feasibility of the solution. Using metaheuristic algorithm of Whale and Genetic, while avoiding any breach of the soft limitations, we managed to come up with a system that provides the most satisfaction between the teachers and students. In our research, we compared Whale and Genetic algorithm with 4 other metaheuristic algorithms. We concluded that the results of Whale and Genetic algorithm are superior to other algorithms in regards to: Improved function goals, less run time, more Pareto front averages, more efficient solutions and results.

Keywords: timetabling, Multi-Objective Model, Increased satisfaction, Whale and Genetic algorithm

Paper Type: Original Research

1. Introduction

Scheduling is a very important aspect of today’s fast pace life. Nowadays, people are constantly trying to be more efficient and get more done in shorter time. The main focus on time management initially was driven to this issue in the late 1950s, after the world war, and during the industrial revolution era. A lot of researchers and mathematicians focused on this matter since. Timetabling is the art of dedicating energy/time to resources in a way that there would be no interaction between two different programs while considering all the limitations. Examples of timetabling charts are: timetabling for school subjects (Post et al., 2011), timetabling for a project (Agarwal et al., 2011), scheduling thecrew(Barrera et al., 2012), public transport scheduling(Shafie et al., 2012), scheduling sports events (Nurmi et al., 2013), and scheduling flights (Pita et al., 2012). Solving the challenges facing the university course timetabling problem (UCTP) is a daunting task due to the diversity among subjects and the broad nature of the topic. There are various approaches to overcome such issues. Based on the time required to solve these problems and their complex nature, these problems are difficult to solve and fall under the category of NP-Hard. These problems are complex and multi-objective, making them challenging to solve. Optimization algorithms are divided into different categories such as exact and approximate algorithms. University timetabling is an NP-hard problem due to its complexity; thus, exact algorithms are not efficient in solving timetabling problems. An approximate algorithm provides close-to-ideal answers in a short time and is capable of solving NP-hard problems, such as the University timetabling problem. Most researchers on this topic agree that timetabling problems are complex

* Corresponding Author: hd_soleimani@yahoo.com

NP-Hard problems that as of yet, no perfect algorithm has been introduced that could solve these problems efficiently. In other words, there is no absolute guarantee algorithms that could solve all these problems. As the problem gets more complex, the time required to run the algorithm is significantly increased as well. Thus, the only solution is utilizing an approximate algorithm that provides us with acceptable results in a reasonable run time. However, the UCTP involves a multi-dimensional scoring system in which faculty and students are linked to courses and classes. Each group's constraints must be considered while planning to design an efficient algorithm. The run time of these algorithms dramatically increase with their scale and complexity (Dokeroglu et al., 2019). Hence, there is no perfect algorithm to solve these problems. In order to overcome this complexity heuristic and metaheuristic algorithms have been developed. In heuristic algorithms, there is a proper rationale behind the solution but that rationale is only applicable to a specific situation and it may not be the best rational. Thus, these heuristic methods are not extensible to other issues and make them less desirable. Another flaw of heuristic algorithms is that they provide one or a few solutions that are supposed to be optimal but in reality, may not be. Basically, they get stuck on local (low level) optimal points and do not explore for better solutions. On the other hand, metaheuristic algorithms are flexible, their results are extensible and can be used to solve variety of problems in the diverse field of the UCTP. Unlike the heuristic algorithms, metaheuristic algorithms do not stop at the local optimal point and constantly utilize random numbers to keep looking for the best and most optimal solution (Garry and Johnson 2002). As a whole, a UCTP involves two categories of objectives; (1) weekly timetabling of the courses, and (2) Examination timetabling. This study, as a contribution, develops a new hybrid metaheuristic algorithm consisting of whale optimization algorithm (WOA) and non-dominated sorting genetic algorithm 2 (NSGA-II) to dedicate courses to lecturers and students during a semester. The satisfaction of both groups must be simultaneously satisfied subject to all the rough constraints. The efficiency of this hybrid method is confirmed by comparing its results with some other metaheuristic algorithms. Besides, this algorithm is successfully conducted in a branch of Islamic Azad University (IAU) in Iran.

2.Literature review

The whale optimization algorithm was first described by Mirjalili and Lewis (2016) from Griffith University in Australia. This algorithm is inspired by observing how humpback whales find their prey in the ocean. They first surround their prey, then go around in circles to make giant bubbles swirling around the hunt to make it confused. This will also help them locate the prey better and increases their chance of success. This method of hunting, gives them multiple advantages. This will allow them to cover a larger area to look for hunting and also a better chance of having a successful hunt. This algorithm can avoid local optimization and achieve global optimization while reducing the number of calculations required. The genetic algorithm is another algorithm inspired by the nature. This algorithm was designed for explaining genetic mutation, populations of chromosomes, and certain parameters such as population size and mutation rate. Natural selection and genetic science were the key components of this algorithm. In this algorithm, chromosomes play the role of answers that are allocated to numbers. The conditions required for a population to evolve correspond to the function of value and quality control on a chromosome in the algorithm. This has been translated into a mathematical function. Natural selection is considered as the fitness function. Similar to DNA gene expression, translation, transcription, and duplication process, here in this algorithm, the date is paired two by two, then split and changed with other data until the final optimized product is produced by process of random selection. In the past, timetabling had to be done manually. People used to sit down, write the number of courses on a piece of paper. Care was taken to not schedule the two most popular courses at the same time. Then other courses would be added to the schedule one by one, from the most to least popular courses. This process would go on until all courses were dedicated to an instructor. This is no longer possible with the number of courses offered at a university each semester. On the other hand, timetabling has university-specific limitations such as the number of students, faculty, and space available so there is no universal protocol. The protocol for each institution has to be developed specifically for that institution. Especially nowadays that more students are enrolled in graduate school, the need for a specific organized system is felt even more. Considering that timetabling is a subcategory of NP-Hard, to date, no specific multi-objective algorithm has been proposed to solve this problem. There is not a definite solution for these types of problems. Various models such as mathematical, computers, graphs, etc. have been proposed in the previous years. All of these models had the same goals: putting in data would provide you with an optimized plan which takes into consideration, goals of the program, needs, and resources. Mathematical models, especially linear models, have a specific application in this field. Timetabling of university courses taking care to have the least deviation from soft limitation has been achieved their goals in a short time compared to other mathematical models and algorithms (Fonseca et al., 2017).

Generally, timetabling includes scheduling courses, classes, and faculty in a fixed period while considering the limitations (Basir et al., 2013). Phillips used an integer programming method to solve the problem of assigning classes in larger scale university settings using an absolute number model (Phillips et. al. 2015). Badoni and colleagues emphasize on the fact that UCTP is a complicated subject with limited resources which makes finding an optimized solution, challenging, to say the least. They proposed an Ant colony optimization algorithm for the university timetabling using events based on groupings of students. This novel idea allocated the students into groups in a way that each student was only member of one group. Then the results were dedicated to different times and rooms based on the solution provided by the Ant colony algorithm (Badoni et al. 2024). Burke et al. (2007) have used a metaheuristic method to provide a solution for timetabling based on graphs. Sabar et al. (2009) has used the honeybee mating algorithm to solve the issue of timetabling for university courses and exams. This nature-inspired algorithm has one big difference to the GA; instead of selecting two parents in each phase, only one parent is needed and the second parent is the queen that is available at all levels. Shokri's group have used a twostep method to solve the timetabling problem at Tehran University. The first step involved using a metaheuristic algorithm to solve the problem, while considering all NP hard limitation, while the second step was to allocate courses to specific classrooms. They used a few neighboring functions to optimize the solution and calculations. This led to an algorithm that generated a solution within a few minutes and had short run time (Shokri et. all, 2023). Metaheuristic approaches have become popular over the past decade. Some examples of such algorithms are firefly algorithm (FA), cuckoo search (CS), artificial bee colon (ABC), Tabu search (TS), genetic algorithm (GA), and simulated annealing (SA) (Badoni et al.,2014). Al-Betar et al (2012) used the memetic algorithm in their research for timetabling. They utilized a harmony search algorithm with a metaheuristic method based on the population. This algorithm was combined with the hill-climbing algorithm to expand their research. It also was combined with the optimum compact particle algorithm to increase the convergence. The results suggested that their method is optimized and is predicted to be useful for complicated data. Some other similar algorithms that have been studied by researchers are great deluge algorithm which is a type of search and Integer Programming tool and logical programming. These have been implemented in artificial intelligence studies (Yadegari et al., 2019). Some articles have looked at optimizing the metaheuristic algorithms such as hybrid algorithm by allocating faculty to periods, classes, and improving timetabling while considering limitations (Abdullah et al., 2010). Murali et al., (2017) have looked into an algorithm based on optimized memetics which uses SA for local probing to solve the problem. Mahiba and Durai (2012) proposed a developed GA using a forbidden search algorithm and considering the problem as a multi-allocating issue. Hiryanto (2013) have looked at solving the dynamic constraint problems using the coloring graph. Mallari's research has proposed a different insight into university timetabling problem by optimizing an approach to course calendars. They designed a multi-objective linear programming model of absolute numbers. Then they assessed the quality of the timetabling calendar bases on the following factors: 1) Compatibility with the schedule of students 2) Being feasible based on the time crunch limitations of students 3) Considering time limitations of the faculty. They achieved good results using an exact algorithm with 100% satisfaction and 9.45 % average optimization based on the timetabling calendars. Subulan and colleagues used a multi-objective optimization model for a capability-based university course timetabling problem. Their algorithm provides a wide range of courses with maximum capacity for the students during a semester. They first categorized the feedback received from students based on the student's understanding of the courses and the course content, then they propose a non-linear absolute number model that considered all hard and soft limitations. This model was successful in utilizing each student's capacities and allocate them to each class in a way that their skills/capacity is best used. They assessed multiple other parameters such as how to allocate each available course to the faculty that has the most skillset in teaching that course. (Subalan et al. 2022). Badoni et al. (2014) Suggested a new algorithm which would form from combining GA and local search algorithm. Alzaqebeh and Abdullah (2015) proposed an ABC algorithm used to solve problems in courses. This algorithm had three unique properties: utilized different methods to choose a self-adjustment system in order to pick the neighboring structure in the search process and was able to combine with other local algorithms. Chang et al. (2017) proposed a new mathematical model to solve the timetabling of the single machine production problem. This model used solid optimization to consider non-definitive transaction's time taking. Unlike the approach of randomized models, this model used an accurate and specific distribution method to process the non-definitive parameters. Algethami et al. used a mathematical model for course timetabling problem with faculty-course assignment constraints. 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). 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.2016). Kong et al. (2017) presented a model for production timetabling of single-machine systems. Pereira and Vásquez (2017) suggested another mathematical model for timetabling a single machine model. They combined branch and bound algorithms and were able to solve problems with up to 300 tasks. Landir et al. (2020) proposed two models of timetabling for high school courses with mixed-integer linear programming. They solved both of these using cooperative parallel methods. They solved one with the cutting plane technique and the other using GA.
Motivated by the relevant literature, there are shortcomings in the context of UCTP that this study tries to cover. In this regard, the main contributions of this study can be highlighted as follows:

- A multi-objective binary integer programming is developed.
- A new hybrid metaheuristic algorithm, consisting of WOA and NSGA-II, is extended.
- A real case study, a branch of IAU, is examined using the proposed model and solution method.

3. Problem definition

A lot of factors should be considered in timetabling for university courses. Some resource limitations to consider are time available to offer the courses during weekdays, hours dedicated to education in a day, and equipment needed for each course. The objective of this study is to present a timetabling model for university courses in a way that maximizes the satisfaction of lecturers and students, while still considering all the limitations. The following rule, as an assumption, for each course should be considered. Each course would be considered 16 hours of education per unit, each education hours is 45 minutes, in a semester. This would be fulfilled within a 6 day/week work schedule in which each day is divided into four time periods from 8 AM to 7 PM. In order to formulate the model, we have to consider some constraints, such as available lecturers, classrooms, time of offering classes, working days, and time blocks in a day. In general, there are three rough constraints that must be satisfied. The first one is resource limitation, for instance, having capable lecturers or having classrooms. The second one is non-overlap, so a teacher only dedicates one course/group/classroom in a given time; the same rule applies to dedicating a classroom to a course/teacher/group and also dedicating a group of students to a teacher/ course in the same given time. The last one is the equipment needed for each course; here, the least needed equipment for each course is the whiteboard or based on the contents of that course could be a video projector, computer, laboratory equipment, recreational equipment, or maps. Moreover, there exist several soft constraints that should be considered. Teacher satisfaction, being able to offer a variety of courses, limiting disruption in the plan, limiting classroom changes, or swapping lecturers between classrooms are examples of soft constraints. The violation of these kinds of constraints leads to the penalty. The assumptions of this study are also listed below:
All sessions of a subject/group should be taught by the same teacher. The minimum and maximum sessions that each faculty has to teach during a semester should be passed; these session numbers differ between faculties. There is no difference in scheduling mandatory versus elective subjects. The minimum required units that a faculty has to offer during a semester is two units and the maximum is nine units per day.

4. Mathematical model

The model is formulated through multi-objective binary integer programming. Each activity includes a lecturer, a course, a class, and a group of students. This way removes unacceptable combinations from the problem by defining each activity based on them. An example of an unacceptable combination is a lecturer or group of students that are not defined for a specific class or course. The final result of this reduces the problem size and the number of variables dramatically.
The sets, parameters, and decision variables are mentioned as follows.

Sets:

$r\epsilon(1,\dots,R)$	the set of classrooms
$d\epsilon(1,\dots,D)$	the set of classes' day
$t\epsilon(1,\dots,T)$	the set of periods

$c\epsilon(1, \dots, C)$	the set of courses
$p\epsilon(1, \dots, P)$	the set of lecturers
$g\epsilon(1, \dots, G)$	the set of students' groups
$k\epsilon(1, \dots, K)$	the set of equipment

Parameters:

W_{cptd}	the satisfaction of lecturer p for course$c$ in the allotted time t on day d
W'_{cgtd}	the satisfaction of a group of students g for courses$c$ in the allotted time t on day d
a_p	the significance of lecturer p for the university
a'_g	the importance of group g of students for the university
h_c	the number of units of course$c$
L_p	the minimum number of units allotted to lecturer p
U_p	the maximum number of unites allotted to lecturer p
H^p_{max}	the maximum number of units allotted to a lecturer p in a day
H^p_{min}	the minimum number of units allotted to a lecturer p in a day
y_{pd}	if lecturer p has the class on the day d will be 1, and otherwise will be 0.
b_{pc}	if lecturer $p$presents the course$c$ will be 1, and otherwise will be 0.
e_{pd}	if lecturer p is available on day d will be 1, and otherwise will be 0.
f_{cd}	if course$c$ is available to be presented on day $d$will be 1, and otherwise will be 0.
ϑ_{rk}	if classroom r has proper equipment $k$will be 1, and otherwise will be 0.
z_{ck}	if course requires equipment $k$will be 1, and otherwise will be 0.

Variables:

x_{cpgtdr}	A binary variable; if lecturer $p$presentscourse c to group of students$g$ in period t of day$d$in classroom $r$will be 1, and otherwise will be 0.

Objective function:

$$Max f_1 = \sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} a_p . W_{cptd} . x_{cpgtdr} \tag{1}$$

$$Max f_2 = \sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} a'_g . W'_{cgtd} . x_{cpgtdr} \tag{2}$$

The first objective function (1) maximizes lecturers' satisfaction with the timetabling. The second objective function (2) maximizes students' satisfaction with the timetabling.

Constraints:

$$\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr} = 1 \ ; \ \forall c \tag{3}$$

Constraint (3) ensures that based on the educational chart, all the courses for a semester should be offered and provided to students.

$$\sum_{g=1}^{G}\sum_{p=1}^{P} x_{cpgtdr} \leq 1 \ ; \ \forall c,d,t,r \tag{4}$$

Constraint (4) guarantees that in each period of each day, only one coursesshould be dedicated to a specific group of students.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{r=1}^{R} x_{cpgtdr} \leq 1 \ ; \ \forall p,t,d \tag{5}$$

Constraint (5) confirms that each lecturer can only offer one courses in each time-period of a day.

$$\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{g=1}^{G} x_{cpgtdr} \leq 1 \ ; \ \forall r,d,t \tag{6}$$

Constraint (6) guarantees that only one course could be offered in a specific classroom at a specific time of the day.

$$\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr} \leq b_{pc} \ ; \ \forall p,c \tag{7}$$

Constraint (7) ensures that acourse will not be assigned to alecturer that does not prefer teaching.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{r=1}^{R} x_{cpgtdr}.h_c \geq L_p \ ; \ \forall p \tag{8}$$

Constraint (8) guarantees that units assigned to each lecturer should be equal to or more than the minimum units assigned to each lecturer.

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

Constraint (9) ensures that units assigned to each lecturer should be equal to or less than the maximum units assigned to each lecturer.

$$\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{r=1}^{R} x_{cpgtdr} \leq 1 \ ; \ \forall g,t,d \tag{10}$$

Constraint (10) ensures that each group of students could only be in one classroom at a specific time in a day.

$$\sum_{p=1}^{P}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{d=1}^{D} x_{cpgtdr}.z_{ck} \leq \vartheta_{rk} \ ; \ \forall c,r,k \tag{11}$$

Constraint (11) ensures that a course requiring special equipment would be held in a class properly equipped.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{r=1}^{R} x_{cpgtdr}.h_c \leq y_{pd}.e_{pd}.H_{max}^{p} \ ; \ \forall p,d \tag{12}$$

Constraint (12) indicates that if lecturer$p$ works on a specific day d he will be assigned maximum of H_{max}^{p}units.

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{T}\sum_{r=1}^{R} x_{cpgtdr}.h_c \geq y_{pd}.H_{min}^{p} \; ; \forall p,d \tag{13}$$

Constraint (13) determines that if lecturer p works on a specific day d he will be assigned minimum of H_{min}^{p}units.

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

Constraint (14) guarantees that course$c$ could only be offered on day d if the chief of educational committee allows.

5. Solution method

Since the studied problem is known as NP-Hard, a novel meta-heuristic algorithm is suggested. As the proposed mathematical model is multi-objective, multi-objective optimization methods should be employed. Thus, the hybrid WOA and NSGA-IImethod is developed. In thewhales' lifestyle, it is preferred to hunt small fishes near thewater surface bycreating bubbles surrounding the prey. TheWOAalgorithm is one of the nature-based and population-based optimization algorithms based on the Whale lifestyle that can be used in a variety of contexts. This algorithm is consisting of sieging, bubble attack, and hunting. These methods have been designed into a mathematical formulation. In WOA, in the sieging operator, different solutions are achieved by increasing the value of a controllable parameter as "a". By selecting random values for vector "A" between −1 and +1, a search agent can be applied. In bubble attach, first it calculates the distance between the wall located in the "X*" and "Y". Then, the position of the whale will be updated to converge walls in the best possible location. Hunting operator is applied by using |A| > 1 mode, while the best solution is chosen when |A| < 1 to update the position of the search agents. The WOA algorithm has the ability to choose between circular or spiral motion. Finally, the WOA algorithmends with satisfying the termination conditions. The whales can identify and surround the hunting grounds. Since the optimal design location in the search space is not known, by comparison, the algorithm assumes that the best candidate for the present is either target hunting or near-optimal. After the best search agent has been identified, other search agents try to update their location to the best search agent (Mirjalili and Lewis 2016; Goli et al. 2020). Distance and position expressions are formulated using Eqs (15) and (16).

$$D = |Q.X^{*}(t) - X(t)| \tag{15}$$

$$X(t+1) = X^{*}(t) - B.D \tag{16}$$

From the above equations, "t" represents the iteration of the algorithm, "Q" and "B" are thecoefficients, and "X*" is the best position obtained and "X(t)" is the current wall position.The value of X* is also updated in each iteration. The values of "Q" and "B" are derived using Eqs. (17) and (18).

$$B = 2.a.r - a \tag{17}$$

$$Q = 2.r \tag{18}$$

Note that"r" is a random vector that takes value between [0, 1].
Furthermore, to have a better search in solution space, crossoverand mutation operators are being used. Besides, since the mathematical modelis multi-objective, to sort the solutions and return best Pareto front, theNGSA-II algorithm is combined with the WOA.

In order to clarify this hybrid algorithm, Fig. 1 depicts its flowchart.

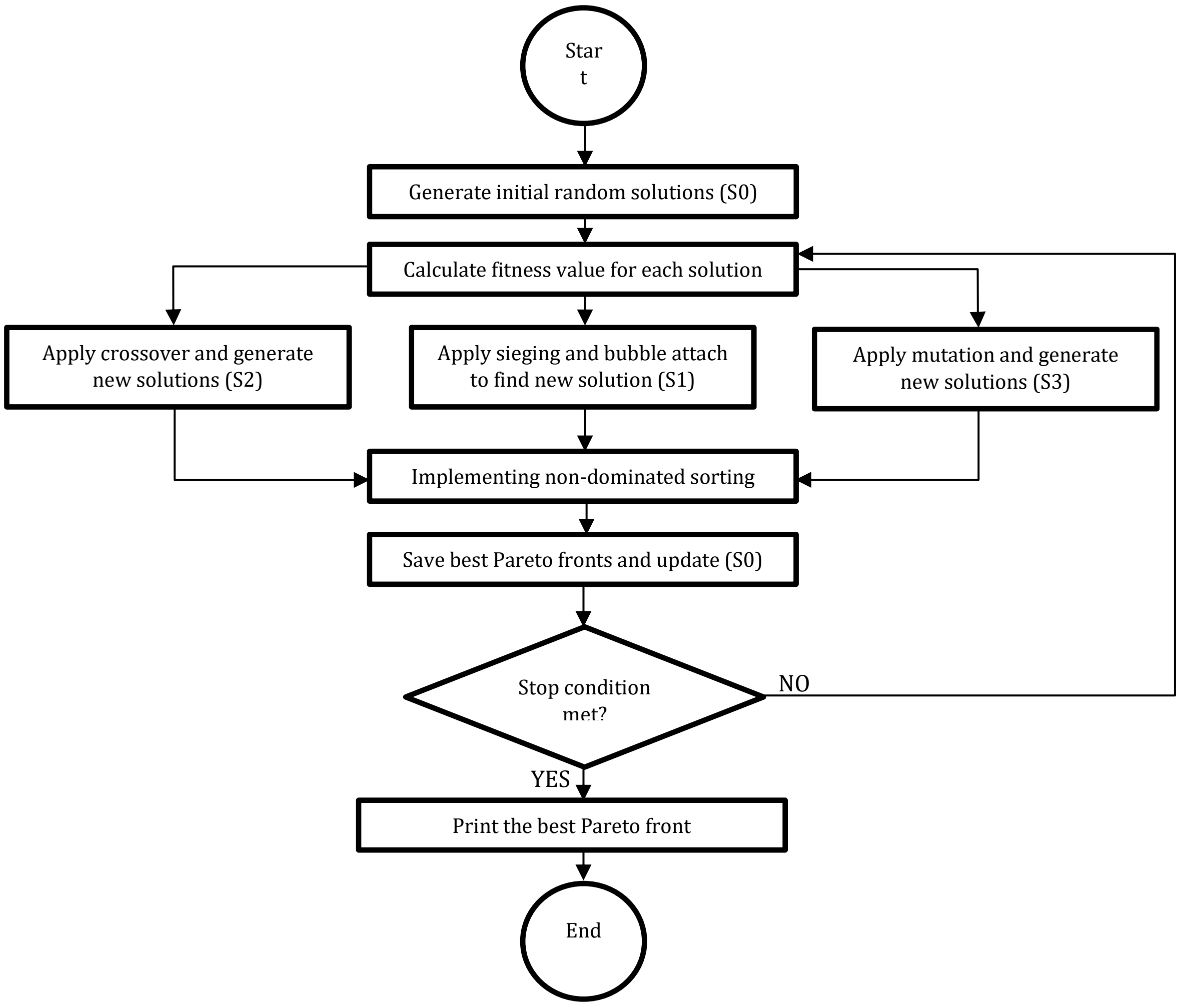
Star
t
Generate initial random solutions (S0)
Calculate fitness value for each solution
Apply crossover and generate new solutions (S2)
Apply sieging and bubble attach to find new solution (S1)
Apply mutation and generate new solutions (S3)
Implementing non-dominated sorting
Save best Pareto fronts and update (S0)
Stop condition met?
NO
YES
Print the best Pareto front
End

Figure 1. The flowchart of the proposed hybrid WOA and NSGA-II

Details of this process are as follow:

1) Start initiates the algorithm
2) Hybrid algorithm will start generating random primary population (consistent numbers between 0-1) for the next repetition.
3) Value of each response will be determined based on function fitness.
4) S1: In this step the whale algorithm will implement its characteristic functions. These functions include encircling prey, creating bubbles surrounding the prey, and swirling around the prey. This algorithm can avoid local optimization and achieve global optimization while reducing the number of calculations required.

S2: Crossover function will exchange random chromosomes from the parent (available responses). Specific differences in chromosome genes will be paid attention to with regard to convergence.

S3: Mutation ad evolution will improve the new generation of chromosomes and makes them more deserving. Then the whale hybrid will procreate and mutation will occur. At this point the cross over part of the genetic algorithm will be replaced by the food seeking behavior of whale algorithm (the mutation will still stay as part of the genetic algorithm). The mutation will occur after procreation of the whale algorithm.

5) Data will be sorted based on the direction of optimization (sorting and dominant perfectionism). Considering that in our research, the procreation will continue 100 times. In each procreation, the next generation whales will be sorted from the previous generation.
6) The most ideal points on the Pareto front (determined based on Topsis method) will be saved and updated. The best sorted whales from the previous step will move on to the next generation.
7) As long as the satisfactory conditions are met, algorithm will be stopped. If satisfactory conditions are not met yet, return to step 3 again.

6.Results

Firstly, the validation of the mathematical modeling is approved through solving by the augmented ε-constraint method. Besides, the results of augmented ε-constraint are compared with the results of WOA-NSGA-II algorithm in order to examine the efficiency of the proposed meta-heuristic algorithm. It should be noted that the augmented ε-constraint method is coded by the GAMS software package. Table 1 indicates the results obtained by both augmented ε-constraint method and WOA-NSGA-II algorithm for five small scale instances. All data, for small scale instances, are generated randomly.

Table 1. The results of augmented ε-constraint and WOA-NSGA-II for small scale instances

Instance	augmented ε-constraint		WOA-NSGA-II	
	OF 1	OF 2	OF 1	OF 2
1	12.3654	7.0021	12.3647	7.0110
2	10.7841	6.9914	10.4412	6.9363
3	12.3002	7.3560	11.9022	7.2139
4	11.3697	6.0051	10.4420	5.8029
5	9.4753	5.5201	9.5443	5.0715

According to Table 1, two important outcomes can be derived. The first one is that the mathematical modeling is valid, and the second one is that the proposed hybrid algorithm indicates a considerable performance to close the global optimal solutions.Now, the proposed hybrid algorithm is tested using a real case study, and its results are compared with four metaheuristic algorithms to evaluate its effectiveness. In the following, the required data from a semester is presented:
This selected branch of IAU has 53 classrooms. Classes are held 6 days a week from Saturday to Thursday. Each day is divided into 150-minute sessions, 135 minutes for the class, and 15 minutes to change classes. One hour for lunchtime is also considered. Thus, 4 periods are considered. The course offered during a semester are 640 course a week which is presented by 219 lecturers, including 33 associate professors, 14 assistant professors, and 5 invitee lecturers from other branches. The classes are divided into 865 class groups. Classroom equipment included computers, an industrial engineering warehouse, laboratories, recreation centers, video projectors, and whiteboards. These are divided into 24 different forms of equipment. The maximum and minimum numbers of units allotted to a lecturer in a day are 1 and 9. The minimum number of units allotted to associate professors, assistant professors, and invitee lecturers are 2, 15, and 16 respectively. Also, the maximum number of units allotted to them are 15, 15, and 30. To evaluate algorithms 40 different runs are conducted for the hybrid WOA-GA, MOFA, MOCS, MOABC, and NSGA-II. Each multi-objective algorithm is tested through its best objective value and the minimum time required to be converged. Plus, mean ideal distance (MID), the number of solutions and Pareto points (No), the maximum spread of diversity (MD), spacing, and time required to run the algorithm is the factor used to compare the performance of the algorithms. All the algorithms are coded by MATLAB 2018a in a PC with an Intel Core i7 processor and 8 GB of RAM. The results are shown in Tables 2-6.

Table 2. The results ofMOCS(Npop10,Maxiter100)

Row	No	MID	MD	Spacing	Min Objs	Max Objs	CPU time (s)
1	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	389.393253
2	6	4.0774	4.614	1.1122	5.4921,5.5734	10.6914,6.63733	380.380296
3	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	67.576192
4	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	363.095719
5	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	86.197433
6	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	87.427226
7	3	2.7873	1.6347	0.49447	9.7217,4.8321	10.7213,6.56012	400.623948
8	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	378.670011
9	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	381.996248
10	4	3.2965	1.5524	0.66155	9.4901,6.3407	10.8263,6.58821	388.257724
11	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	383.897925
12	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	67.394513
13	5	4.4695	2.0975	0.69355	8.9306,6.1848	10.1304,7.34639	364.225606
14	3	3.4505	1.4627	0.47734	9.8328,5.8146	10.1869,7.19569	372.510159
15	4	3.3386	2.5232	0.89747	8.2823,5.8396	10.7437,6.30606	367.175125
16	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	374.632475
17	3	3.1526	1.4521	0.49995	9.3597,5.2993	9.803,6.6217	373.797456
18	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	495.77886
19	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	372.386388
20	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	386.266934
21	3	4.4857	1.7376	0.49928	8.3075,6.3384	10.4623,6.67999	378.374545
22	3	3.4505	1.4627	0.47734	9.8328,5.8146	10.1869,7.19569	381.344274
23	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	381.145216
24	3	3.7186	2.1084	0.57048	8.3705,6.0552	10.8797,6.49637	379.693525
25	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	435.93543
26	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	377.280925
27	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	362.990215
28	3	4.381	2.2562	0.81377	8.9004,6.2806	11.3158,7.45174	376.923551
29	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	367.715977
30	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	370.641711
31	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	360.575269
32	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	368.741979
33	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	417.169821
34	3	2.8457	1.6096	0.49226	9.9842,5.4259	11.383,6.72893	59.637626
35	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	386.856856
36	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	369.021079
37	4	4.8013	2.0952	0.65617	8.2864,6.1495	10.4203,6.85692	381.182537
38	3	5.0446	2.0603	0.40945	8.4195,5.961	10.743,7.15195	377.535353
39	4	3.3386	2.5232	0.89747	8.2823,5.8396	10.7437,6.30606	384.166365
40	4	3.005	1.8616	0.61355	9.5201,5.2458	10.6248,6.66725	366.43718

Table 3. The results of MOABC

Row	No	MID	MD	Spacing	Min Objs	Max Objs	CPU time (s)
1	4	5.4175	2.0127	0.66577	6.1937,4.97	8.2104,5.3709	313.6319
2	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	243.7762
3	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	238.6636
4	2	12.108	1.7153	0	6.0457,5.5746	8.9866,5.6608	330.3337
5	3	12.304	2.03	0.49886	6.5483,4.953	8.8454,6.2509	289.3191
6	2	4.7807	1.1914	0	7.728,4.6578	7.3844,6.0349	299.8714
7	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	237.3566
8	2	3.8733	1.0321	0	7.6152,5.5497	8.6732,5.6735	245.4813
9	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	235.7737
10	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	244.3067

11	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	234.6121
12	2	3.2261	1.3764	0	7.995,5.6	9.8892,5.662	274.1609
13	2	3.3889	1.0455	0	8.3409,4.9227	8.4808,6.0068	310.9488
14	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	241.9006
15	2	5.565	1.2487	0	6.5583,5.3901	8.1138,5.4967	280.3671
16	3	3.5288	1.83	0.51297	7.4733,4.5978	9.0416,5.8339	248.377
17	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	373.0182
18	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	230.7345
19	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	297.7795
20	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	235.7886
21	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	297.3477
22	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	265.543
23	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	233.851
24	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	217.4256
25	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	237.2939
26	2	3.1845	1.0636	0	8.5055,5.0231	7.3763,5.0919	292.5777
27	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	233.9039
28	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	240.8496
29	2	3.3889	1.0455	0	8.4808,4.9227	8.3409,6.0068	311.4655
30	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	242.0116
31	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	216.6561
32	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	233.6411
33	2	3.3889	1.0455	0	8.4808,4.9227	8.3409,6.0068	306.9043
34	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	232.0533
35	2	2.8371	0.80157	0	8.5495,5.315	9.1761,5.4571	243.8937
36	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	242.3563
37	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	236.2458
38	2	3.9818	0.65718	0	7.5451,5.3692	7.9252,5.5743	293.8685
39	5	6.1223	3.6139	0.73828	5.755,5.1076	10.2547,6.38378	238.8366
40	5	4.4098	2.129	0.73161	7.0002,4.6831	8.6273,5.7906	232.64

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

Row	No	MID	MD	Spacing	Min Objs	Max Objs	CPU time (s)
1	8	3.7158	3.6152	0.78854	8.4924,5.6234	10.9611,6.85459	478.3974
2	7	2.7228	2.6953	0.86332	9.7195,5.256	11.511,6.58945	396.8362
3	8	4.1402	3.2477	0.82385	8.1063,5.4678	10.5197,6.70001	483.2539
4	3	3.89	1.3015	0.82139	9.7434,6.8037	11.1888,6.80373	389.3041
5	7	2.6806	2.9288	0.86358	9.7115,5.5276	11.8037,6.56167	497.0369
6	5	3.8012	2.1437	0.74522	9.3627,5.3916	10.5554,7.19523	386.8295
7	6	5.2937	3.3713	0.95368	7.7962,5.9478	12.0411,6.74946	424.1873
8	2	2.9086	0.93163	0	10.0881,5.73926	10.0997,6.60713	486.0346
9	5	5.0091	1.8823	0.72841	7.8743,5.6112	9.3752,5.6112	467.0851
10	8	6.5467	4.7633	1.09	7.3424,5.0653	11.814,6.89804	517.4983
11	8	3.6288	3.1479	0.86031	8.2652,5.5039	10.518,6.53402	487.9683
12	2	4.012	0.54021	0	0.10908,0.59666	0.40016,0.080525	504.6668
13	8	4.9261	3.5983	0.93137	7.3327,5.4733	10.4768,6.37052	537.3668
14	5	3.2689	2.0615	0.67891	9.112,4.6933	10.0936,6.55319	134.7113
15	6	4.3617	2.959	0.75506	8.9515,5.8996	11.0811,7.57221	157.8631
16	8	2.9801	3.6917	0.84561	9.2614,4.6669	11.6789,6.97048	127.1581
17	5	3.2924	2.3995	0.90467	9.0104,4.9115	9.2905,6.8022	142.2533
18	2	4.012	0.54021	0	0.10908,0.59666	0.40016,0.080525	161.3064
19	8	3.4286	3.1299	0.81467	7.9853,4.5069	9.4878,6.275	182.8087
20	7	3.3858	3.1496	0.93371	8.5129,5.6041	10.2657,6.78724	190.212
21	8	2.4543	3.767	0.79701	9.8439,4.3662	11.4563,6.96977	142.1386
22	6	4.1887	3.3873	0.83072	8.037,5.9246	10.8427,6.89954	141.1436
23	7	3.2715	2.6113	0.87605	9.089,5.4495	11.19,6.60674	153.9007
24	8	4.578	3.5567	0.84388	7.9837,5.3771	10.6529,6.86254	149.1644
25	7	9.973	3.4303	0.82565	7.0906,5.4904	10.3134,6.83224	142.9151
26	8	3.6119	3.3918	0.78746	8.7824,5.5778	10.9265,7.06433	379.1483
27	5	2.753	1.7036	0.85056	10.3844,5.3051	11.1058,6.6443	174.0149
28	7	7.6521	3.7863	0.79786	7.468,4.5839	10.0588,7.1488	106.7101
29	8	4.3525	3.4763	0.82238	8.1638,5.9365	10.5736,6.92236	148.837
30	8	3.6288	3.1479	0.86031	8.2652,5.5039	10.518,6.53402	155.8771
31	8	3.6119	3.3918	0.78746	8.7824,5.5778	10.9265,7.06433	164.4036
32	2	2.3451	0.68895	0	11.4154,5.97043	11.6074,6.40451	166.3039
33	7	4.9827	2.9376	0.84486	8.5895,6.142	10.3891,7.41904	149.8101
34	8	3.098	3.5278	0.87739	9.1926,5.8573	11.6708,6.90287	143.7992
35	6	5.0074	2.8977	0.74684	7.3109,5.7037	9.732,6.4294	172.1561
36	8	3.4925	2.9981	0.78397	8.8357,5.2249	10.4355,6.81337	157.6984
37	7	2.6321	2.7765	0.84599	8.9985,5.2895	11.0445,5.86303	148.521

38	8	3.7673	2.6101	0.90407	9.4847,5.7893	10.6037,7.3191	140.7252
39	5	2.8209	2.2569	0.72703	9.4793,5.628	11.0865,6.42384	160.0226
40	6	3.2661	1.8854	0.99347	101984,6.25934	11.6831,7.02162	155.1001

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

Row	No	MID	MD	Spacing	Min Objs	Max Objs	CPU time (s)
1	6	3.6297	2.3986	0.77255	8.8221,5.469	10.468,6.69537	267.0957
2	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	237.8897
3	5	3.391	2.1665	0.86032	9.9962,5.3578	10.7906,7.39146	252.3823
4	7	16.0888	3.8328	0.85107	5.9392,5.6395	10.1077,7.41005	245.0695
5	6	3.2371	3.7244	0.78799	9.0366,5.311	10.3654,6.3309	197.0864
6	7	2.935	2.8852	0.81116	9.8125,5.359	11.6404,6.99803	210.5561
7	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	189.8917
8	5	4.1015	2.5362	0.74556	8.422,6.269	11.2025,6.77113	213.4704
9	6	4.4439	3.8968	0.77636	7.5828,5.3655	12.3226,6.72705	196.0851
10	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	185.1998
11	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	234.7775
12	6	2.8936	1.6485	0.72279	10.664,6.36713	10.3654,6.3309	304.0194
13	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	190.8236
14	6	5.5912	3.7882	0.77765	7.7609,5.8515	10.3654,6.3309	274.7113
15	4	7.4373	2.6886	0.64809	7.0484,6.0356	10.0677,6.53241	217.8631
16	5	5.8652	2.5387	0.74632	7.5319,5.7651	10.3406,6.65241	367.1581
17	6	2.8936	1.6485	0.72279	10.664,6.36713	11.3328,6.89602	282.2533
18	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.3654,6.3309	281.3064
19	8	3.8018	3.5799	0.79243	8.7077,4.861	10.7445,7.26843	232.8087
20	6	5.1547	3.3873	0.79894	7.8203,5.817	11.3681,6.97566	240.212
21	6	4.5308	3.8622	0.77555	8.2265,5.561	10.3654,6.3309	262.1386
22	6	21.077	3.5561	0.78867	7.2995,5.4479	10.3654,6.3309	191.1436
23	6	2.2777	2.8134	0.88923	7.8203,5.817	11.3681,6.97566	323.9007
24	7	6.4592	4.564	0.8562	5.6175,5.3944	10.4446,7.18135	199.1644
25	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	322.9151
26	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	359.5838
27	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.5885,6.97081	246.0149
28	6	3.2761	2.9969	0.77808	8.1045,4.7514	10.3654,6.3309	340.6662
29	7	4.6879	3.7751	0.92378	7.3946,5.9659	11.3059,6.5856	238.837
30	6	4.0077	3.5917	0.7201	8.4707,4.6429	11.9317,7.20338	295.8771
31	6	21.077	3.5561	0.78867	7.2995,5.4479	10.6692,7.20104	204.4036
32	6	3.3512	3.8905	0.85921	7.8537,5.6416	11.2213,6.47552	316.3039
33	7	6.4592	4.564	0.8562	5.6175,5.3947	10.4446,7.18135	237.8101
34	6	5.5912	3.7882	0.77765	7.7609,5.8515	11.9039,7.13208	293.7992
35	6	5.4452	2.7528	0.94042	8.0627,6.1289	10.3654,6.3309	232.1561
36	4	3.8595	2.2698	0.61121	9.0961,5.748	10.9224,7.27629	287.6984
37	6	2.8936	1.6485	0.72279	10.664,6.36713	10.3654,6.3309	298.521
38	5	3.5989	1.9087	0.69614	8.9175,5.9949	10.1635,6.60344	340.7252
39	8	4.248	3.8987	0.94461	8.1551,6.0233	10.9129,7.04679	220.0226
40	5	3.391	2.1665	0.86032	9.9962,5.3578	10.7906,7.39146	205.1001

Table 6. The results of WOA-NSGA-II (Npop10, Maxiter100)

Row	No	MID	MD	Spacing	Min Objs	Max Objs	CPU time (s)
1	9	2.999	4.2525	0.83419	8.6862,4.9117	11.7396,6.68615	206.3975
2	8	3.9656	3.6548	0.7709	7.9892,4.8698	10.3165,6.74114	271.7431
3	9	5.7183	4.2336	0.85814	8.0453,5.4859	11.3236,7.45402	97.7277
4	8	3.9656	3.6548	0.7709	7.9892,4.8698	10.3165,6.74114	258.4376
5	9	7.7748	5.2741	0.87768	6.8779,4.7298	12.2229,7.30193	359.1859
6	8	2.9457	2.7335	0.83555	9.7615,5.4107	11.2118,6.79905	254.4057
7	9	2.8093	4.9821	0.99818	8.8797,5.5213	12.8253,7.09139	130.9776
8	8	3.9656	3.6548	0.7709	7.9892,4.8698	10.3165,6.74114	228.5368
9	8	3.7872	3.4891	0.78751	8.4617,5.3928	10.9694,6.85889	371.3242
10	9	2.999	4.2525	0.83419	8.6862,4.9117	11.7396,6.68615	210.528
11	9	4.8439	4.079	0.86069	7.9025,5.5442	11.307,6.96046	323.1548
12	8	3.8159	3.7428	0.84714	8.5709,5.7397	11.8683,7.1589	208.2727
13	9	3.8364	4.5673	0.84549	7.5374,4.7495	11.2211,6.55575	180.7658
14	8	10.369	4.1176	0.81555	6.9913,4.9607	10.6695,6.78935	366.5846
15	8	3.8159	3.7428	0.84714	8.5709,5.7397	11.8683,7.1589	199.0616
16	9	7.7748	5.2741	0.87768	6.8779,4.7298	12.2229,7.30193	358.7594
17	9	10.472	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	265.9133
18	9	3.0567	3.7709	1.0003	9.1416,5.571	11.4033,6.84329	260.2047
19	8	10.369	4.1176	0.81555	6.9913,4.9607	10.6695,6.78935	63.3337
20	9	3.5174	3.5436	0.8507	8.9452,5.8784	10.9468,7.14449	202.5844
21	9	3.8364	4.5673	0.84549	7.5374,4.7495	11.2211,6.55575	184.8945
22	9	3.8364	4.5673	0.84549	7.5374,4.7495	11.2211,6.55575	178.1566
23	9	10.472	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	315.7819
24	9	5.7183	5.2741	0.87768	6.8779,4.7298	12.2229,7.30193	361.7455
25	9	10.472	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	298.0442
26	9	7.7748	5.2741	0.87768	6.8779,4.7298	12.2229,7.30193	368.9092

27	8	3.9656	3.6548	0.7709	7.9892,4.8698	10.3165,6.74114	264.2928
28	8	3.7872	3.4891	0.78751	8.4617,5.3928	10.9694,6.85889	368.446
29	9	10.472	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	316.1113
30	8	3.8159	3.7428	0.84714	8.5709,5.7397	11.8683,7.1589	199.6867
31	9	4.0708	3.8502	0.80074	8.4556,5.5189	11.0596,7.16075	176.1479
32	8	4.6156	3.5319	0.81279	7.8756,5.6095	10.3798,6.83998	290.0322
33	9	4.0307	3.8386	0.85162	8.4728,6.0434	11.7675,6.94465	372.2921
34	9	10.472	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	279.1514
35	9	5.0712	4.6152	0.83314	7.5497,5.4878	11.348,6.93052	214.6033
36	8	10.025	3.9784	0.82543	6.801.5.0825	10.4478,7.27647	122.142
37	9	7.7748	4.7737	0.91185	6.6866,6.0339	11.4379,6.86742	285.0839
38	9	4.8439	4.079	0.86069	7.9025,5.5442	11.307,6.96046	332.4619
39	8	3.9656	3.6548	0.7709	7.9892,4.8698	10.3165,6.74114	235.9232
40	9	3.5226	4.1528	0.86669	8.7702,4.9771	11.516,7.24573	47.769651

Interpretation of the results:

According to Table 2, Theaverage Maximum Objectives (optimized point on Pareto front) are 10.65218, the average minimum number of goal function is 7.78878, and average maximum number of second goal function is 6.823258. Average minimum second goal function is 5.25601, average minimum and maximum number (points on the Pareto front) of the first function is 9.77003 and the same number for the second function is 6.289632. Average time required to do 40 runs of the Cuckoo algorithm is 344. 1263 seconds. The number of Pareto points is 3-6 (3.65 average), the average distance compared to ideal is 3.896348, the most distribution number of the points in this algorithm averages at 1.995903. The average distance numbers of this algorithm are 0.598702. Based on the results in Table 3: Theaverage maximum number of first objective function is 8.97655, and the average minimum number of first objective function is 6.99504. The average maximum number for the second objective function is 5.887878 and the average minimum number of second objective function is 5.01025. The average maximum and minimum number of Pareto front in the first function is 7.985797 and this was 5.449064 for the second Pareto front numbers. The average run time (for the 40 runs) of honeybee colony was 261.3892 seconds. The number of Pareto fronts were between 2-5 (3.75 is average). The average distance from the ideal in this algorithm is 5.568798. The highest distribution (scattering) number for the points in this algorithm averages at 2.067499. The mean Spacing is 0.445993. Based on Table 4 data: Theaverage Maximum number for the first objective is 10.25949, while the average minimum number for the first objective function is 7.759005. Theaverage Maximum number for the second objective is 6.418554, while the average minimum number for the second objective function is 5.29339. The average maximum and minimum number of Pareto front in the first function is 9.308267 and this was 5.8323324 for the second Pareto front numbers. The average run time (for the 40 runs) of Genetic algorithm was 260.1292 seconds. The number of Pareto fronts were between 2-8 (6.375 is average). The average distance from the ideal in this algorithm is 3.987323. The highest distribution (scattering) number for the points in this algorithm averages at 2.758198. The mean Spacing is 0.755132. Based on Table 5 data: The average Maximum number for the first objective is 10.79905 while the average minimum number for the first objective function is 7.0104. The average Maximum number for the second objective is 6.87386, while the average minimum number for the second objective function is 5.3269. The average maximum and minimum number of Pareto front in the first function is 9.379025 and this was 6.133629 for the second Pareto front numbers. The average run time (for the 40 runs) of Firefly algorithm was 255.936 seconds. The number of Pareto fronts were between 4-8 (6.15is average). The average distance from the ideal in this algorithm is 5.798708. The highest distribution (scattering) number for the points in this algorithm averages at 3.231473. The mean Spacing is 0.777608. Based on Table 6 data: The average Maximum number for the first objective is 11.29926, while the average minimum number for the first objective function is 7.84315. The average Maximum number for the second objective is 6.93955, while the average minimum number for the second objective function is 5.33492. The average maximum and minimum number of Pareto front in the first function is 9.605298 and this was 6.134329 for the second Pareto front numbers. The average run time (for the 40 runs) of Whale and Genetic hybrid algorithm was 2507394 seconds. The number of Pareto fronts were between 8-9 (8.625 is average). The average distance from the ideal in this algorithm is 5.635975. The highest distribution (scattering) number for the points in this algorithm averages at 4.201225. The mean Spacing is 0.796847.

1. **Interpretation of the multi -objective model:**

Considering that both aim functions are at a maximum, the output of aim function possesses a better satisfaction rate in the mathematical model. We ran the minimum and maximums of all 5 algorithms and compared them. Based on the data extrapolated from the tables after doing 200 runs, the Whale and Genetic algorithm, carries the best aim function numbers and the most satisfaction rate compared to all other algorithms.

2. **Interpretation of run times for all multi-objective algorithms:**

What matters in timetabling for university courses is solving the mathematical model using the most out of university resources and considering all limitations in the shortest run time possible. Again, the Whale and Genetic algorithm had the shortest run time and most efficiency between all algorithms.

3. **Interpretation of number of number of solutions provided by thePareto front for the metaheuristic algorithm:**

   Knowing that the final answer of any multi-objective model will result in a group of points on the Pareto front, and an algorithm that could generate more diverse Pareto fronts is superior to the other algorithms, we noticed that Whale and Genetic algorithm has the most Pareto front points and is a better algorithm because it provides the user with more choices and reliable results. The provided answers on the Whale and Genetic algorithm were more efficient than the other algorithms.

4. **Interpreting mean ideal distance:**

This reflects the average Pareto answers compared to an ideal point of each algorithm. The least mean ideal distance, the better that Pareto front. Mean ideal distance is also an element of convergence of algorithms and is extrapolated from the following formula: $MID = \frac{1}{n}\sum_{i=1}^{n} c_i$

The whale and Genetic algorithm had the second best mean ideal distance (after Cuckoo algorithm) between all.

5. **Interpreting maximum spread or diversity results:**

We utilized this as a method to compare algorithm in our study. Maximum spread defines the most diversity of Pareto front answers and is calculated using the difference between starting points and ending points of the Pareto front from the following formula: $MD = \sqrt{\sum_{m=1}^{M}\left(\max_{i=1:|Q|} f_m^i - \min_{i=1:|Q|} f_m^i\right)^2}$

Whale and Genetic algorithm, had the best maximum spread compared to all other algorithms.

6. **Interpretation of Pareto front point spacing results:**

This category was also utilized in our study to compare algorithms. Spacing is calculating the Euclidean distance on the Pareto front which means calculating the shortest distance between two different points. This is calculated using Pythagoras non negative ratio and Manhattan distance (organized and equal distance) with the following formula: $D_{euc} = (\sum_{i=1}^{p} x_i - y_i)^2)^{\frac{1}{2}}$

$$D_{man} = \sum_{i=1}^{p} |x_i - y_i|$$

We utilized MATLAB software and Function distance to come up with the results.

The whale and Genetic algorithm overall had the best results compared to all other algorithm in most categories.

Moreover, Figure 2indicates six different runsof the hybrid WOA and NSGA-II.

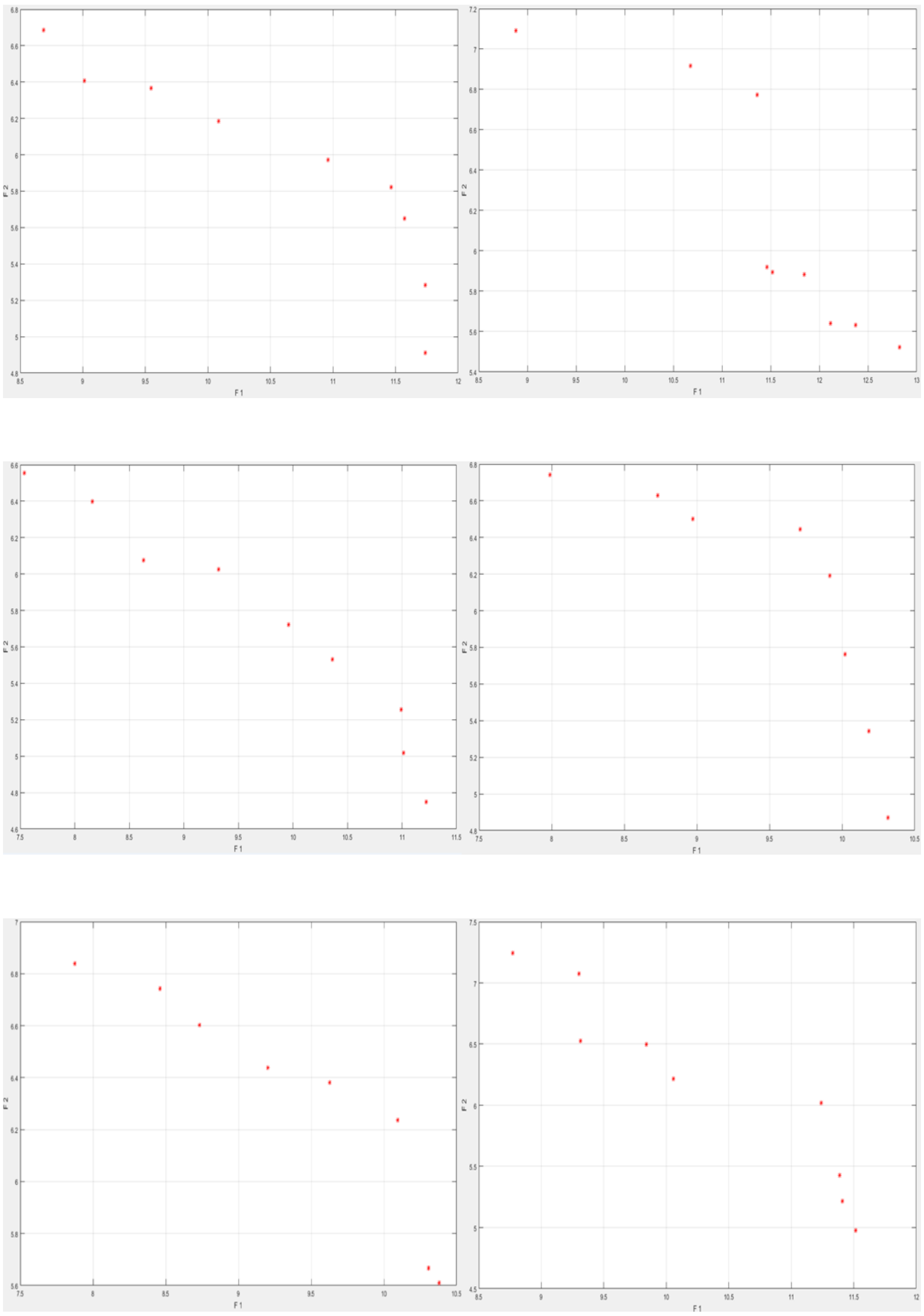

Figure. 2. Pareto surface for six different runs of the hybridWOA and NSGA-II

Figure 2 depicts the Pareto front points of 6 out of 40 different runs (numbers 1,7,13,27,32,40). The algorithm was able to reach these points in a fast and efficient time. These points represent the maximum optimization points between all 7 algorithms. Using Topsis method, the most optimized point will be picked from all the available Pareto front points on each chart.

7.Summary and concluding remarks

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 semester’s timetable or trial and error is used which has a lot of shortcomings 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. This number was allocated based on the knowledge base, experience, publications, and the specific courses that each faculty taught. 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 (200 total runs) were performed and results were compared between the algorithms. We concluded that the Whale and Genetic algorithm is superior to all other algorithms in reference to maximum and minimum number of optimized aim functions, 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. This paper addressed a multi-objective binary integer programming model in which the satisfaction of lecturers and students was considered as two conflicting objectives. A new hybrid algorithm based on WOA and NSGA-II was developed. This algorithm was compared with four multi-objective metaheuristic algorithms in terms of No, MD, MID, spacing, and CPU time. The results indicated that the hybrid WOA-NSGA-II could outperform the other ones due to the quality of results and running time. The proposed algorithm could find the most Pareto solutions while presenting the best values of the maximum spread of diversity and spacing. Implementation of this algorithm in a real case study led to maximizing the satisfaction of both lecturers and students at the same time. Although this study provided some interesting managerial implications, there were several limitations. For instance, all the parameters of this study were static; however, some of them, such as provided units in a semester, would be dynamics. Similarly, the satisfaction of lecturers and students were considered as a definite number. Future studies can extend the present study through several aspects. One of the major aspects is to represent heuristic or metaheuristic algorithms, and then compare their results with this study’s findings. In a similar way, bi-level relaxation and decomposition methods can be implemented. Likewise, the mathematical model can be developed through practical assumptions; the model can be reformulated under uncertainty through fuzzy or robust optimization models.

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.

Al-Betar, M. A., Khader, A. T., & Zaman, M. (2012). University course timetabling using a hybrid harmony search metaheuristic algorithm. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42 (5), 664-681.

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.

Badoni, R. P., Kumar, S., Mann, M., Mohanty, R. P., & Sarangi, A. (2024). Ant colony optimization algorithm for the university course timetabling problem using events based on groupings of students. In Modeling and Applications in Operations Research (pp. 1-36). CRC Press.

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.

Burke, EK., McCollum, B., Meisels, A., Petrovic, S. & Qu, R., (2007), A Graph-Based Hyper-Heuristic for Educational Timetabling Problems, European Journal of Operational Research, Vol: 176(1), 177-192.

Chang, Z., Wu, H., Pan, K., Zhu, H., & Chen, J. (2017). Clean production pathways for regional power-generation system under emission constraints: A case study of Shanghai, China. Journal of Cleaner Production, 143, 989-1000.

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering, 137, 106040.

Fonseca, G. H., Santos, H. G., Carrano, E. G., & Stidsen, T. J. (2017). Integer programming techniques for educational timetabling. European Journal of Operational Research, 262(1), 28-39.

Garey, M. R., & Johnson, D. S. (2002). Computers and Intractability, vol. 29.

Goh, Say Leng, Graham Kendall, and Nasser R. Sabar. Improved local search approaches to
solve the post enrolment course timetabling problem. European Journal of Operational research.

Goli, A., Tirkolaee, E. B., & Weber, G. W. (2020). A perishable product sustainable supply chain network design problem with lead time and customer satisfaction using a hybrid whale-genetic algorithm. In Logistics Operations and Management for Recycling and Reuse (pp. 99-124). Springer, Berlin, Heidelberg.

Hiryanto, L. (2013, June). Incorporating dynamic constraint matching into vertex-based graph coloring approach for university course timetabling problem. In 2013 International Conference on QiR (pp. 68-72). IEEE.

Kong, L., Li, H., Luo, H., Ding, L., Luo, X., &Skitmore, M. (2017). Optimal single-machine batch scheduling for the manufacture, transportation and JIT assembly of precast construction with changeover costs within due dates. Automation in Construction, 81, 34-43.

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.

Mahiba, A. A., & Durai, C. A. D. (2012). Genetic algorithm with search bank strategies for university course timetabling problem. Procedia Engineering, 38(1), 253-263.

Mallari, C. B., San Juan, J. L., & Li, R. (2023). The university coursework timetabling problem: An optimization approach to synchronizing course calendars. Computers & Industrial Engineering, 184, 109561.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

Murali, G. B., Deepak, B. B. V. L., Bahubalendruni, M. R., & Biswal, B. B. (2017). Optimal assembly sequence planning using hybridized immune-simulated annealing technique. Materials Today: Proceedings, 4(8), 8313-8322.

Nurmi, K., Goossens, D., &Kyngäs, J. (2013). Scheduling a triple round robin tournament with minitournaments for the Finnish national youth ice hockey league. Journal of the Operational Research Society, 65 (11), 1770-1779.

Pereira, J., & Vásquez, Ó. C. (2017). The single machine weighted mean squared deviation problem. European Journal of Operational Research, 261(2), 515-529.

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.

Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2012). "A honey-bee mating optimization algorithm for educational timetabling problems". European Journal of Operational Research, 216(3), 533-543 .

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.

Shokri, A., Dana, T., Hemmasi, A., & Toutounchian, S. (2023). Investigating the relationship between occupational stress and job satisfaction among university faculty members (case study: one university in Tehran). Innovation management and operational strategies, 4(1), 73-81.

Subulan,K. E. M. A. L., & Gürsaç, A(2022).A Multiple Objective Optimization Model for a Novel Capability-Based University Course Timetabling Problem: A Case Study at Deu Industrial Engineering Department.

TavakoliM.M.,SHIROUYEHZAD H.,HOSSEINZADEH LOTFI F.,NAJAFI E(2018),Proposing A New Mathematical Model For Planning Of University Course Timetabling Based On Quality Of LessonPresentation, Journal of operational research and its applications, 15(3),45-66.

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.

Werra, D.de, (1996), Practice and Theory of Automated Timetabling, Springer Berlin Heidelberg, Vol: 1153, 296-308.

Yadegari, E., Alem-Tabriz, A., & Zandieh, M. (2019). A memetic algorithm with a novel neighborhood search and modified solution representation for closed-loop supply chain network design. Computers & Industrial Engineering, 128, 418-436.

Appendix:

The designed code is a consistent code based on a fitness model:

```
c=1:nc
 tc=0;
if tc==1;break;end
NVAR(1)=ng*np*nc;
```

For example, if we assume 3 groups, 7 professors and 2 courses, the 2 following Matrix will be utilized to allocate the courses to the professors (data from one semester were input into Excell MATLAB program and used for this)

```
>>x1=rand(3,7,2)
x1=(:,:,1)
0.1484   0.0700   0.0715   0.3942   0.1287   0.0602   0.0379
0.0805   0.5246   0.6813   0.6093   0.1978   0.07038  0.1196
0.7115   0.7982   0.4286   0.9963   0.8461   0.5827   0.3951
```

```
x1=(:,:,2)
0.5672   0.9138   0.1778   0.9643   0.7384   0.7937   0.7462
0.2674   0.3675   0.3438   0.5874   0.1692   0.5271   0.6381
0.3928   0.5736   0.8517   0.8974   0.0072   0.2310   0.8649
a1(1,:)=x1(g,:,c);a1=a1.*b(:,c)';%L7

a1=a1.*EP;
 [vpc,up]=sort(a1,'descend');up(vpc==0)=[];
for p=up
    if tc==1;break;end
NVAR(2)=ng*np*nc*nd;.
a2(1,:)=x2(g,p,c,:);a2=a2.*f(c,:);%L14.
[vd,ud]=sort(a2,'descend');ud(vd==0)=[];
NVAR(3)=ng*np*nc*nd*nk*nr;
for d=ud
   if tc==1;break;end

   k=find(z(c,:)==1);
  a3(1,:)=x3(g,p,c,d,k,:);a3=a3.*v(:,k)';%L11
  for t=1:nt

s4(:,:)=X(c,:,:,t,d,r);%L4
if sum(s4(:))>0;continue;end

S5(:,:,:)=X(:,p,:,t,d,:);%L5
if sum(s5(:))>0;continue;end

S6(:,:,:)=X(:,:,:,t,d,r);%L6
if sum(s6(:))>0;continue;end


S10(:,:,:)=X(:,:,g,t,d,:);%L10
if sum(s10(:))>0;continue;end

if VP(p)+h(c)>U(p);continue;end%L9

if VPD(p,d)+h(c)>Hmax(p);continue;end%L12

X(c,p,g,t,d,r)=1;%L3

YP(p)=1;

Z1=Z1+(W(c,p,t,d)*a(p));%L1

Z2=Z2+(WP(c,g,t,d)*ap(g));%L2

VP(p)=VP(p)+h(c);
VPD(p,d)=VPD(p,d)+h(c);
if VP(p)>U(p);EP(p)=0;end
if VPD(p,d)>Hmax(p);EPD(p,d)=0;end
tc=1;break

if tc==0
CH(1)=CH(1)+1;

p=1:np
if YP(p)==0;continue;end
if L(p)>VP(p)
CH(2)=CH(2)+L(p)-VP(p);%L8

for d=1:nd
if Y(p,d)==0;continue;end
if Hmin(p)>VPD(p,d)
CH(3)=CH(3)+Hmin(p)-VPD(p,d);%L13
SCH=sum(CH);
sol.fit=fit'*(1+SCH);
sol.SCH=SCH;
fit=[1/Z1 1/Z2];
fit(isnan(fit))=10^10;
sol.info.RealFit=[Z1 Z2]
```