Generating Optimal Timetabling for Lecturers using Hybrid Fuzzy and Clustering Algorithms

Hamed Babaei [1*], Amin Hadidi[2]
[1]Department of Computer Engineering, Islamic Azad University,Ahar Branch, Ahar, Iran,
hamedbabaei63@gmail.com
[2] Department of Mechanical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran,
amin.hadidi@yahoo.com, a-hadidi@iau-ahar.ac.ir

Abstract

University course timetabling is an NP-hard problem which must be performed for each semester frequently. The major technique in the presented approach would be analyzing data to resolve uncertainties of lecturers' preferences and constraints within a department in order to obtain a ranking for each lecturer based on their requirements within a department where it is attempted to increase their satisfaction and develop lecturers timetabling by using clustering algorithms. The first goal of this paper is to improve satisfaction of lecturers and then optimize the ranking of lecturers based on soft constraints weights over their preferences. The proposed method applies a two-step algorithm. At the first step, the department performs timetabling process using a fuzzy decision making approach to prioritize and rank lecturers by local search algorithm with seven neighbor structures and genetic algorithm to improve lecturers' ranks as well as thoroughly satisfying hard constraints over the department in a local manner. In the second step, two clustering and traversing agents are used, where the former clusters lecturers of the department and the latter finds the extra resources. Following the clustering and traversing, in order to reach the major goals of the paper, mapping action is performed based on lecturers' constraints in resources. In this method, the list of each lecturer's selective preferences is resolved, prioritized and ranked by applying a fuzzy decision making method based on fuzzy comparison of daily and weekly timeslots of per lecturer and then the timetable including department lecturers with their fitness functions is given to the hybrid algorithm in order to improve the quality of fitness function of lecturers within each timetable, so that the clustering and mapping is performed based on a desired logic of each lecturer's fitness function. The applied datasets are in terms of satisfying the scheduling requirements in the real world of computer engineering department of Islamic Azad University of Ahar branch.

Keywords: clustering algorithms, fuzzy multi criteria-decision making approach, hybrid approach, university courses timetabling

1. Introduction

UCTTP (University Course Timetabling Problem) is an important problem in the universities performing at each semester frequently, known as an exhausting and time-consuming task. On the other hand, this problem is in the class of those problems with non-polynomial time complexity. So, in order to avoid this repetitive and time consuming process, we must find facilitator procedures for this problem which is the major motivation of this research. Furthermore, UCTTP is a hybrid optimization problem that leads to some issue to solve this problem optimally and analytically. It means that non-polynomial time complexity of UCTTP problem is due to increasing problem size with the growing number of students requiring computations with exponential complexity. UCTTP problem performs the allocation of all events (courses, teachers and students) to a number of timeslots and classrooms within one semester so that no conflict arises in such allocations. This problem also mustsatisfy both hard and soft constraints during allocation of events to resources, so that the possible timetables are obtained after full satisfaction of whole hard constraints and also timetables with high quality after satisfying a maximum number of soft constraints and it is not necessary that soft constraints are satisfied completely as hard constraints [1, 2, 3, 4, 5, 6].

The presented goals to solve UCTTP problem in this research include:

1- Descending constraints and preferences satisfaction of lecturers-departments,
2- Improving the configuration of soft constraint weights by using fuzzy values,
3- Increasing lecturers' satisfaction by performing local search method

These goals are evaluated by hybrid fuzzy comparison and clustering of lecturers and grouping the extra resources of the department.

2. Related Works

In 2010, a two-part graph edges coloring method [7] has been presented to solve UCTTP problem where this method has been tested on datasets of three semesters and their analysis was based on the comparison of all penalties within a predefined set of violated soft constraints. The aim of this method was to reduce the number of penalties and to create high quality timetables over the manually generated timetables. Here, a two-part graph consists of a graph with vertices within two sets X and Y separately and each edge in the graph has an end-point in another set. In 2008, an ant colony optimization algorithm [8] has been used to solve UCTTP problem after registration by using international competitions timetabling 2007. Ants allocate events to the rooms and timeslots based on two types of pheromone T_{ij}^{s} andT_{jk}^{y}. This type of pheromone represents the probabilities of allocating an event i to timeslot j and room k. This algorithm shows good performance on timetabling and produces better results during long execution. In 2012, a multi-population hybrid genetic algorithm [9] has been proposed to solve UCTTP problem. As genetic algorithm has multi direction search feature, so it is useful to solve this type of problems as an efficient method. In this paper, three types of genetic algorithms of FGARI, FGASA and FGATS are proposed. In the proposed algorithm, fuzzy logic is used to evaluate the number of violations from soft constraints in the fitness function in order to deal with real world data that are ambiguous and unreal. However, random methods, local search, simulated annealing and Tabu search are used accompanied with fuzzy method to improve the inductive search in order to satisfy search capability and also not individually using genetic algorithm to avoid entrapment in local optimality.

A paper [10], presents a new single-parameter local search heuristic named step counting hill climbing algorithm (SCHC). It is a very simple method in which the current cost serves as an acceptance bound for a number of consecutive steps. This is the only parameter in the method that should be set up by the user. Furthermore, the counting of steps can be organized in different ways;

therefore, the proposed method can generate a large number of variants and extensions. They investigate the behavior of the three basic variants of SCHC on the university exam timetabling problem. However, their new method has two additional advantages: a more flexible acceptance condition and better overall performance. In paper [11], the first step in the successful solution of this problem was to define a course structure model that allowed application of classical course timetabling methods. Several methods were necessary to solve the complete problem. First, support procedures were needed to detect and correct an infeasible problem where hard constraints were being violated. These problems are described formally using a weighted constraint satisfaction model of the timetabling problem and solutions are proposed through two types of the algorithms: (1) generic iterative forward search with conflict based statistics, and (2) branch and bound.

Student sectioning is the problem of assigning students to particular sections of courses they request while respecting constraints such as course structures, section limits, and reserved spaces. [12], Students may also provide preferences on class times and course alternatives. In this paper, three approaches to this problem are examined and combined in order to tackle it on a practical level: student sectioning during course timetabling, batch sectioning after a complete timetable is developed, and online sectioning for making additional changes to student schedules. The post-enrolment course timetabling (PE-CTT) is one of the most studied timetabling problems, for which many instances and results are available. This work [13] designs a Meta heuristic approach based on simulated annealing to solve the PE-CTT. They consider all the different variants of the problem that have been proposed in the literature and they perform a comprehensive experimental analysis on all the available public instances. Another paper [14], proposes a hybrid local search algorithm for the solution of the Curriculum-Based Course Timetabling Problem and we undertake a systematic statistical study of the relative influence of the relevant features on the performances of the algorithm. In particular, they apply modern statistical techniques for the design and analysis of experiments, such as nearly orthogonal space-filling Latin hyper-cubes and response surface methods. As a result of this analysis, their technique, properly tuned, compares favorably with the best known ones for this problem.

Many real-life problems are dynamic, with changes in the problem definition occurring after a solution to the initial formulation has been reached. [15], a new iterative forward search algorithm is proposed to solve minimal perturbation problems. Significant improvements to the solution quality are achieved by including new conflict-based statistics in this algorithm. The proposed methods were applied to find a new solution to an existing large-scale class timetabling problem at Purdue University, incorporating the initial solution and additional input changes. The problem consists of assigning courses to teaching terms and years, satisfying a set of precedence constraints and balancing students' load among terms.

Differently from the original Generalized Balanced Academic Curriculum Problem (GBACP) formulation, in [16], the same course can be assigned to different years for different curricula (i.e., the predetermined sets of courses from which a student can choose), leading to a more complex solution space. The problem is tackled by both Integer Programming (IP) methods and combinations of meta-heuristics based on local search.

A large number of variants of the timetabling problem have been proposed in the literature by [17], which differ from each other based on the type of institution involved (university or school) and the type of constraints. This problem that has been traditionally considered in the operational research field has recently been tackled with techniques belonging also to Artificial Intelligence (e.g., genetic algorithms, tabu search, and constraint satisfaction).

In particular, [18] focuses attention on the formulation known as the curriculum-based course timetabling problem (CB-CTT), which has been tackled by many researchers and for which there are many available benchmarks. The contribution of this paper is twofold. First, they propose an effective and robust single-stage simulated annealing method for solving the problem. Second, they design and apply an extensive and statistically principled methodology for the parameter tuning procedure. The outcome of this analysis is a methodology for modeling the relationship between search method parameters and instance features that allows us to set the parameters for unseen instances on the basis of a simple inspection of the instance itself. The case [19], describes a number of improvements that have been made to the course timetabling solver used in the open source university timetabling system UniTime since their last paper on this topic (Rudova et al, 2011). This progress is demonstrated on benchmark data sets from Purdue University that were introduced in the earlier paper and that are available online.

3. The Proposed Method

In the proposed algorithm, two phases are determined to achieve the goals of paper [20].

3-1- First phase of the proposed method

The proposed algorithm consists of three steps:

1. Generating the initial possible timetables by complete satisfaction of hard constraints related to available lecturers and resources within a department (coordinator unit),
2. Ranking and prioritizing lecturers’ soft constraints and available resources by applying fuzzy decision making algorithm (the first part of optimization unit), and
3. Optimizing the quality of timetables generated in the second step by using a hybrid algorithm (the second part of the optimization unit), as shown in Fig. 1.

In this algorithm, the process of scheduling lecturers in the available resources of a department is designed in three units as the following: The first unit is the user interface which provides the communication of users with the server’s database and coordinator unit. The second unit performs the process of generating initial timetables based on non-violation of lecturers’ hard constraints and available resources within a department.

The third unit consists of two parts where the first part prioritizes and ranks soft constraints of lecturers and available resources within a department by applying fuzzy multi criteria decision making algorithm over soft constraints and to increase the satisfaction of each timetable, the second part uses a hybrid algorithm in order to improve the quality of soft constraints within the timetables ranked in the first part.

The first unit performs timetabling and required coordination among department's server database and coordinator unit by communicating with the system users including lecturers, students and training users of each department. This unit provides a mutual relation between users and system.

In the second unit (coordinator unit), the hard constraints of lecturers and available resources of each department are satisfied as following:

1. A lecturer could not teach more than 6 hours per day.
2. A student or a group of students could not be in more than one classroom at the same timeslot, simultaneously.
3. A lecturer could not be in more than one department at the same timeslot in one day, simultaneously.
4. Two lecturers could not be in the same classroom at the same time.
5. When a course is allocated to one classroom, that class must provide the required facilities, features and capacity of that course.
6. A lecturer could not teach for more than one student group within a department, simultaneously.
7. The number of allocated courses to each lecturer and maximum number of units corresponding to each course must be determined per lecturer.

The third unit consists of two sections. Initially, in the first section (optimization unit), the fuzzy multi criteria decision making method is ranked based on soft constraints in a department in terms of priorities and values of each constraint so that at each solution a list of soft constraints could be satisfied with a descending sequence based on their weights which would be according to fuzzy decision making and uncertainty of mentioned constraints of lecturers and available resources. In the second section (optimization unit) optimizing the quality of timetables generated in the second step by using a hybrid algorithm.

Internal structure of timetabling system as shown in Fig. 1

1. Courses database with the related characteristics of each course in terms of course type
2. Lecturers database
3. Classrooms database (classrooms and related features of each one)
4. Daily and weekly timeslots database
5. Student groups database
6. Coordinator unit (in addition to examining the hard constraints of a department, this unit performs coordination process among all existing units within the scheduling system and the output of this unit is formatted as a solution(s) with the format of understandable timetables for optimization unit in order to apply the given soft constraints in the department).

7. Fuzzy decision-making methods could be studied in the first part of optimization unit for prioritizing and ordering lecturers based on their features and constraints.
8. Metaheuristic approach (hybrid metaheuristic approaches could be used by connecting to the coordinator unit).

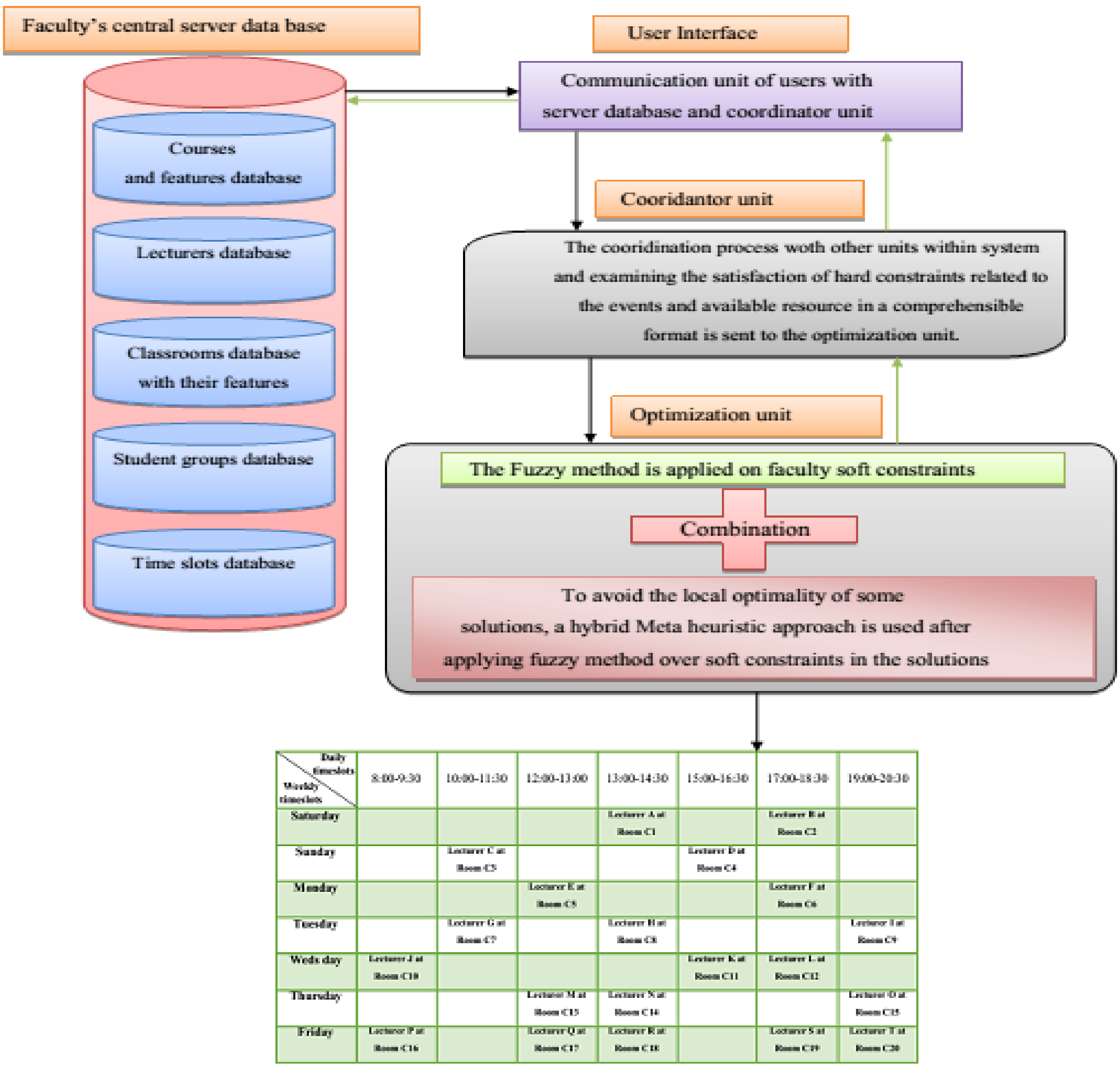


Fig. 1. General view of research schematic flowchart (phase 1)

In the second step, the process of applying fuzzy multi criteria decision-making approach is performed to rank and prioritize lecturers based on equations (rules) (1), (2), (3) and (4). The process of ranking and prioritizing these events depends on the features of the daily timeslots. At first, lecturers address their weekly timeslot selections and then by presenting their demands and preferences, their ranks are determined in terms of daily timeslots selections per day of the week, so that lecturers' ranking would be performed more precisely in terms of their daily timeslot selections and then the effect of this selection and ranking in daily timeslots would result in a prioritization through weekly timeslots for lecturers. At the end of this section, when the prioritization and ranking of lecturers based on their daily/weekly timeslot selections have been finished as fuzzy multi criteria comparison, now it is time to adjust lecturers' prioritization and ranks based on their selections and features.

The methodology is that at first, lecturers present their daily timeslot selections as a fuzzy list which is different from faculty selections for events. The comparison is performed based on their features according to equations (1), (2), (3) and (4). [21].

$$T\left(\widetilde{E_l} \geq \widetilde{E_{l+1}}\right) = \max_{x \geq y}\left\{min\left(\mu_{\widetilde{E_l}}(x), \mu_{\tilde{E}_{l+1}}(x)\right)\right\} \tag{1}$$

$$T\left(\widetilde{E_l} \geq \widetilde{E_{l+1}}, \widetilde{E_{l+2}}\right) = min\{T\left(\widetilde{E_l} \geq \widetilde{E_{l+1}}\right), T\left(\widetilde{E_l} \geq \widetilde{E_{l+2}}\right)\} \tag{2}$$

$$T\left(\widetilde{E_l} \geq \widetilde{E_l}\right) = T\left(\widetilde{E_l} \leq \widetilde{E_l}\right) = T\left(\widetilde{E_l} = \widetilde{E_l}\right) = 1 \tag{3}$$

$$T\left(\widetilde{E_l} > \widetilde{E_l}\right) = T\left(\widetilde{E_l} < \widetilde{E_l}\right) = 0 \tag{4}$$

In equation (1), the way of fuzzy comparison of two events $\widetilde{E_\iota}$ and $\widetilde{E_{\iota+1}}$ is as the following. At first, we obtain a fuzzy list in terms of events' priorities and then perform detailed comparison for each set of events done mutually. It means that at first, $T\left(\widetilde{E_\iota} \geq \widetilde{E_{\iota+1}}\right)$ and then $T\left(\widetilde{E_{\iota+1}} \geq \widetilde{E_\iota}\right)$ is compared and evaluated. The comparison process is performed over the existing events' features and priorities within the department where we could obtain all comparisons of an event over others by reaching the comparative state of an event. In equation (2), all three fuzzy comparison rules$T\left(\widetilde{E_{\iota+1}} \geq \widetilde{E_\iota}\right)$, $T\left(\widetilde{E_{\iota+1}} \geq \widetilde{E_\iota}\right)$ and $T\left(\widetilde{E_{\iota+1}} \geq \widetilde{E_\iota}\right)$ must be determined so that lately the comparison of events are finished according to right side of equation 2. Equations (3) and (4) are used as reminder relations for both equations (1) and (2).

When the timetables of the second phases are entered into the third phase, the amount of violations that events have from their features must be calculated as equation (5):

$$f(Sol) = \sum_{i=1}^{w} f_i \cdot w_i \tag{5}$$

(Here, let *w*, as the number of soft constraint weights, $f_i$as violation amount of each soft constraint and $w_i$as the weight of each soft constraint).

The third phase includes two parts, where the first part is to apply local search algorithm with random iteration and the second part consists of a genetic algorithm for $f(Sol)$ which has been improved by the filter of local search algorithm with random iteration and it would be improved further as much as possible by the genetic algorithm.

Before entering into the first part of the third step, the hybrid neighboring structures with the required random iteration are stated in terms of *f(Sol)*s to improve each *Sol* as the following:

N_1: random selection of a lecturer and exchanging the timeslot of two courses related to that lecturer, so that the hard constraints of the lecturer and course would not be violated.

N_2: random selection of a course and transmitting it to other timeslots randomly

N_3: random selection of a course and lecturer change, even if it is necessary to change the timeslot and classroom

N_4: random selection of a course and then selecting a course with the same unit number (holding duration) and course subject, so that the timeslots of those courses must be exchanged with each other.

N_5: transmitting an event (lecturer or course) from one timeslot to different timeslots

N_6: switching two events (lecturer or course) in two timeslots

N_7: changing two events (lecturer or course) in three separate timeslots by using N_5 and N_6

When *Sol*s have been passed in the first part of step 3 by local search algorithm with random iteration to minimize the violations of lecturers from their features, now we reach to the second part of step 3, namely applying genetic algorithm to further improve the timetables obtained from the output of the local search algorithm with random iteration. The steps of the genetic algorithm include the following steps. At first, the number of initial population or timetables obtained by local search algorithm with the improvement of random iteration is determined. In this algorithm, each chromosome represents a timetable or a lecturer with classrooms, timeslots, courses and selective priorities. The structure of gene at each chromosome (timetables) consists of lecturer code, course code, daily timeslot, weekly timeslot and selective priority for each course by the lecturer. As the timetables within the population have become chromosomes (lecturers), so the violations from the selective priorities corresponding to each lecturer must be evaluated per chromosome by equation (5). The genetic algorithm starts from this step that for each generation, which is better to be one to three generations, the following steps must be performed to generate timetables with the capability of scheduling in one semester.

1- The roulette wheel is used to select chromosomes for crossover process. This wheel generates a random number in [0, 1] to select each chromosome. Then, the chromosome is selected corresponding to the range of that random number. Here, the probability of selecting a chromosome would be based on fitness function or *f (sol)* of each timetable. So chromosomes with higher fitness get higher probability than other chromosomes with lower fitness. However, this selection method allows the chromosomes with lower probability to have a chance to be selected. The selection of a chromosome and its probability are studied. If the fitness of chromosome is higher than random number R, then that chromosome is placed in the pool of crossover, otherwise the fitness value of that chromosome is replaced with the fitness value of next chromosome (it could be selected either sequentially or randomly) until the cumulative sum of *f (sol)* values of the selective chromosomes become larger or equal to the generated random number R and then a parent which has led to increase the f (s*ol)* values of the chromosomes over the generated R (random number) is transmitted to the crossover pool and this is done until the parent selection is taken place as the number of initial populations (for crossover and mutation).

2- Following the chromosome selection process, the crossover operation must be applied to generate child chromosomes in order to replace the chromosomes with lower fitness function value or f(sol) to be transmitted to the next generation. There are different types of crossover operators which could be used according to the type of problem. To perform crossover operation, we use the method of common genes of two chromosomes (a timetable with lecturer) according to the content and structure of each gene and data type of selective priorities of each lecturer per course. However, we perform this to generate all children (lecturers) of each generation. It means that the crossover of two chromosomes would be based on the replacement of each lecturer's selective preferences for each course, if daily and weekly timeslots are the same for that gene (commonly). After applying crossover operator, in order to avoid randomness of replacements and transmission of events (lecturers) within resources, we would not use mutation operator.

3-2- Second phase of the proposed method

There are four agents in the proposed algorithm, shown in Fig. 2. 1- Timetable per department or agent TA, 2- interface (MA), 3- clustering agent (CA) and 4- traversing agent (TraA). In this algorithm, lecturers timetabling process is designed in three phases as the following [22, 23, 24]. The first part consists of steps 1 and 2 planned by timetabling agent (TA) to generate feasible and non-conflict timetables. The second part consists of steps 4, 5 and 6 that performs the process of clustering lecturers to make uniform distribution over the traversed resources of the department by TraA. The third part includes steps 7 and 8 that performs the process of mapping lecturers' clusters in resources based on their constraints and transmits timetables with planning capability to the department for one semester. The first part includes hard constraints related to lecturers and resources which have been satisfied by department.

Across parts 1 and 2, the interface agent (MA) examines the extraction operations of lecturers with their features in order to cluster them as improved and non-conflict in the next step based on the major goal, namely timetabling of lecturers and sending them to their corresponding department to modify and eliminate the conflict when it detects a conflict or inconsistency in timetables of department lecturers. Then the timetables of the lecturers that have been stabilized by interface agent, are sent to clustering agent (CA) during step 3.

In the second part, CA clusters department lecturers based on their constraints (K-means [22], fuzzy C-means [23] and funnel clustering [24] algorithms) (step 4) and TraA is applied by traversing and grouping additional resources of department (step 5). However, before entering into step 5, all occupied and additional resources must be determined from department timetables in step 6 and sent to step 5 to perform traversing and grouping process by TraA.

In the third part, the mapping process of preferences, demands and requirements of lecturers are presented to uniformly distribute and allocate among department resources. In the last step of phase 3 (step 8), the final solution (the timetable of department lecturers for one semester) is sent to the target department based on the identification codes of department after mapping CA clusters to the traversed resources of TraA agent.

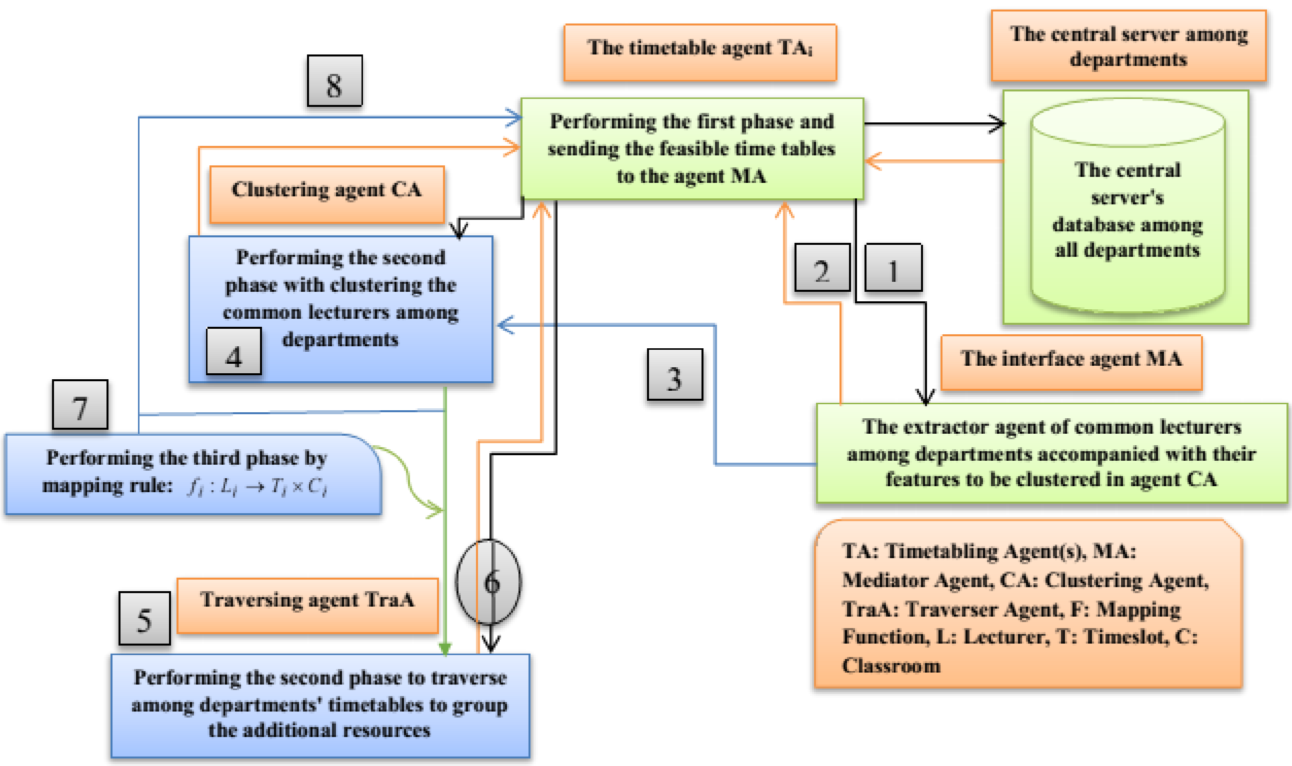


Fig. 2. General view of research schematic flowchart (phase 2)

A pseudo code of proposed algorithm's structure in Fig. 3 is presented, formally andbriefly. At the end of section 3, the framework of the proposed algorithm is presented as a pseudo code.

The proposed algorithm: Generating Optimal Timetabling for Lecturers using Hybrid Fuzzy and Clustering Algorithms

Input: events (lecturers, L_i) and resources (timeslots and classrooms, R_i) of each department

Begin

Phase 1:

Step 1: Generating an empty time table and allocate the priorities and demands of lecturers within a department to produce the initial solutions of lecturers

Step 2: Ranking and prioritization in respect of satisfying soft constraints per lecturer in feasible time tables generated in phase 1 accompanied with satisfying hard constraints related to each lecturer (fuzzy multi criteria decision making).

Step 3: Improving the quality of time tables generated based on satisfying soft constraints in the objective function of phase 2 through applying a hybrid meta-heuristic method as local search and genetic algorithms.

Phase 2:

Step 1: Timetabling agents (input: events (lecturers) and resources, output: timetables) // TA_i agents

Step 2: Mediator agent (input: timetables, output: timetables including lecturers among departments) // MA agent

Step 3: Clustering agent (input: timetables received from MA and clustering process, output: clustered lecturers, L_i) // CA agent

- **3-1:** Applying k-means algorithm
- **3-2:** Applying fuzzy c-means algorithm
- **3-3:** Applying proposed funnel clustering algorithm

Step 4: Traverser agent (input: timetables of each department, output: the traversed resources per department, R_i) // TraA agent

Step 5: Mapping (input: clustered lecturers, L_i and traversed resources, R_i, output: mapping of f_i: $L_i \rightarrow T_i \times C_i$ per department)

End

Output: The planned timetables for each department to inform lecturers timetabling

Fig. 3. Pseudo-code of the proposed algorithm's

4. Experimental Results and Comparisons

To test the proposed algorithm, the dataset includes lecturers, computer engineering department, daily/weekly timeslots and classrooms of Islamic Azad University, Ahar Branch where there are 30 lecturers, 1 department, 7 weekly timeslots (Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday), 7 daily timeslots (each day starts with 8:00-9:30 timeslot and other timeslots are as 10:00-11:30, 12-13, 13-14:30, 15-16:30, 17:00-18:30, the last timeslot would be 19-20:30) and we consider 13 classrooms (3 practical classes and 10 theoretical classes).

Equations 6 and 7 are applied to evaluate the satisfaction of lecturers within each cluster and among clusters based on their preferences and related constraints, respectively. Here, $CTDS_1^{(i)}$ calculates the percent of descending satisfaction of each lecturer's features at each cluster and $CTDS_2^{(i_j)}$also represents the percent of descending satisfaction of lecturers' preferences and features among clusters and over each cluster. Equation (6) is formulated inter clusters as the following [22, 23, 24].

$$CTDS_1^{(i)} = \frac{\sum_{k=1}^{n} W_{ik}^{SC} \times X_{ik}}{\sum_{const=1}^{r}\left(Total_{const}^{SC}\right)} \times 100 = \frac{\sum_{Const=1}^{r}\sum_{k=1}^{n}\left(W_{ik}^{Const} \times L_{ik}\right)}{\sum_{Const=1}^{r}\sum_{k=1}^{n} Total_{k}^{Const}} \times 100 \qquad (6)$$

In equation (6), i[th] cluster represents k lecturers $k = 1, \ldots, n$ and W_{ik}^{SC} is a constraint(s) for lecturer X_{ik} (k[th] lecturer in i[th] cluster) satisfied byW_{ik}^{SC}. In this equation,$Total_{Const}^{SC}$represents all constraints of lecturers within each cluster per lecturer. Equation (7) is presented as intra clusters (out of clusters) as the following.

$$CTDS_2^{(i_j)} = \frac{\sum_{i=1}^{c}\left(W_i^{SC} \times i\right)}{\sum_{j=i+1}^{c}\left(Total_{Const_j}^{SC} \times j\right)} \times 100 = \frac{\sum_{Const=1}^{r}\sum_{k=1}^{n}\sum_{i=1}^{c} W_{ik}^{Const} \times i}{\sum_{Const=1}^{r}\sum_{j=i+1}^{c} Total_{Const_j}^{Const} \times j} \times 100 \qquad (7)$$

In equation (7), $i = 1, \ldots, c$ is the number of clusters, W_i^{SC} is the satisfaction percent of lecturers' constraints of i[th] cluster and j represents the number of other clusters in addition to i[th] cluster wherej = i + 1,…, c. Here, the value of W_i^{SC} must be calculated by the ratio of the number of the satisfied constraint(s) for x[th] lecturer in i[th] cluster to all constraints of i[th] cluster for the existing lecturers within that cluster.

Figs.4, 5, 6, 7 and 8 represent the results obtained by applying algorithms over the lecturers' constraints. In Fig.4, fuzzy multi criteria decision-making algorithm is shown by applying clustering algorithms.

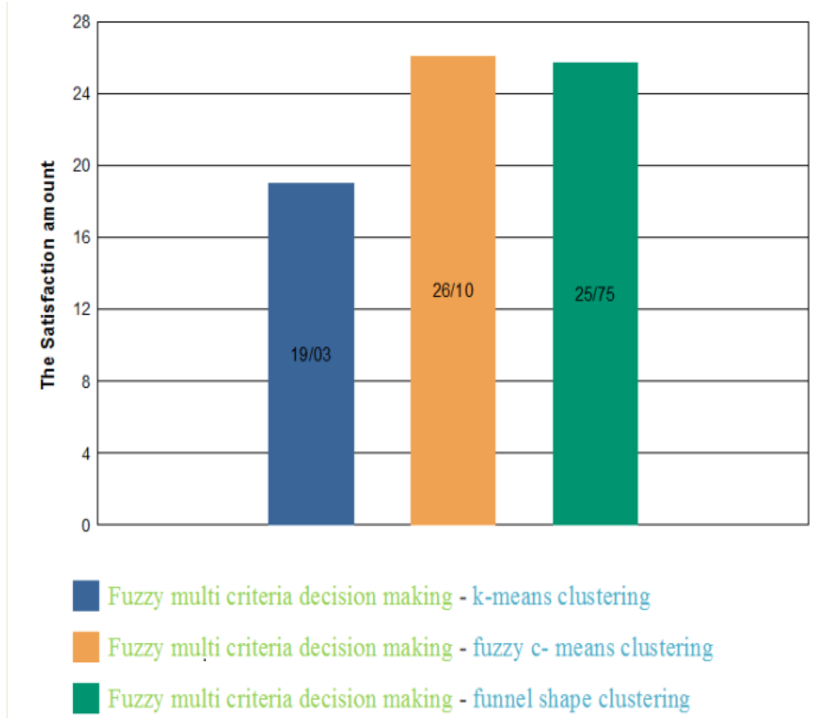


Fig.4. The result of applying fuzzy multi criteria decision making algorithm with clustering algorithms

In Fig.5, hybrid fuzzy multi criteria algorithms (local search) with clustering techniques are shown,

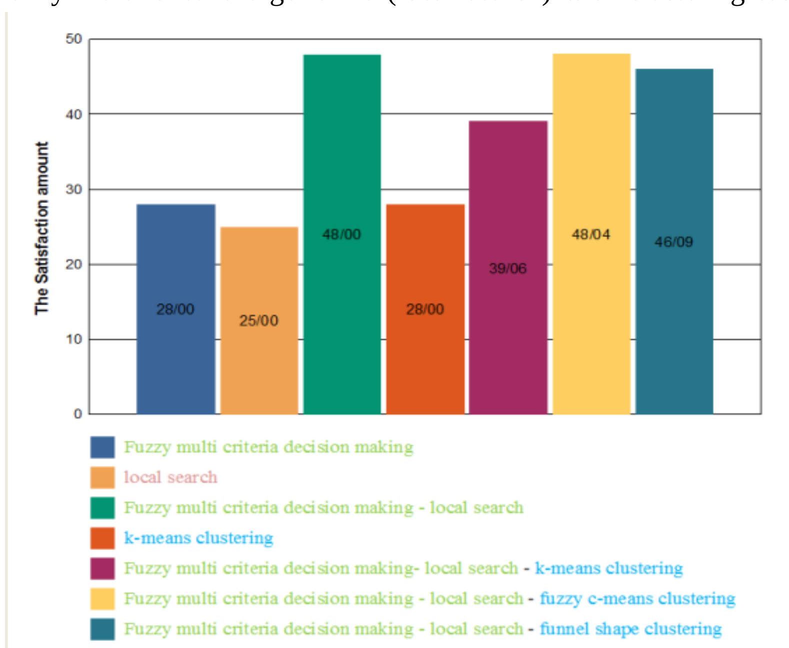


Fig.5. The result of applying hybrid fuzzy multi criteria algorithms (local search) with clustering algorithms

In Fig.6, local search algorithm is presented with clustering techniques.

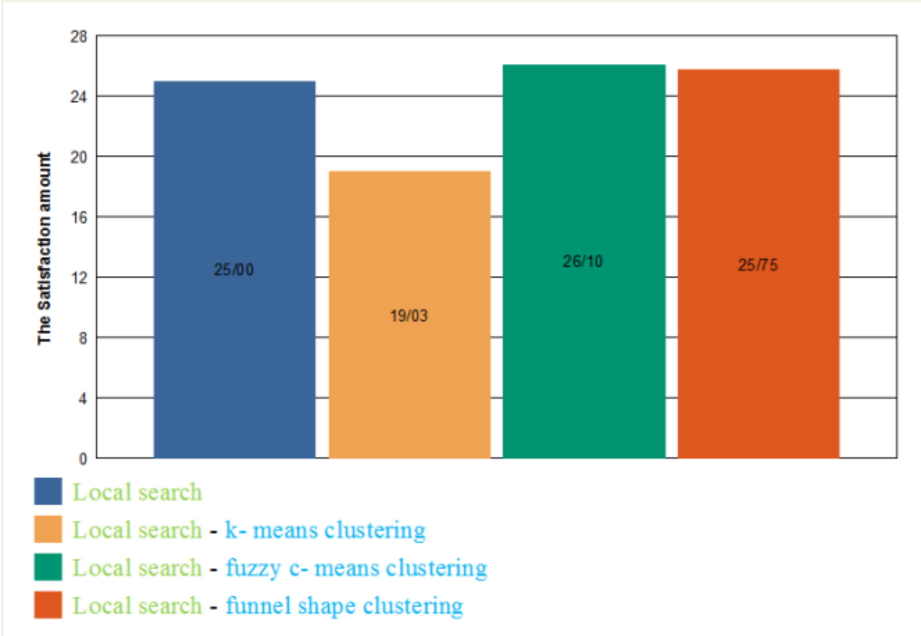
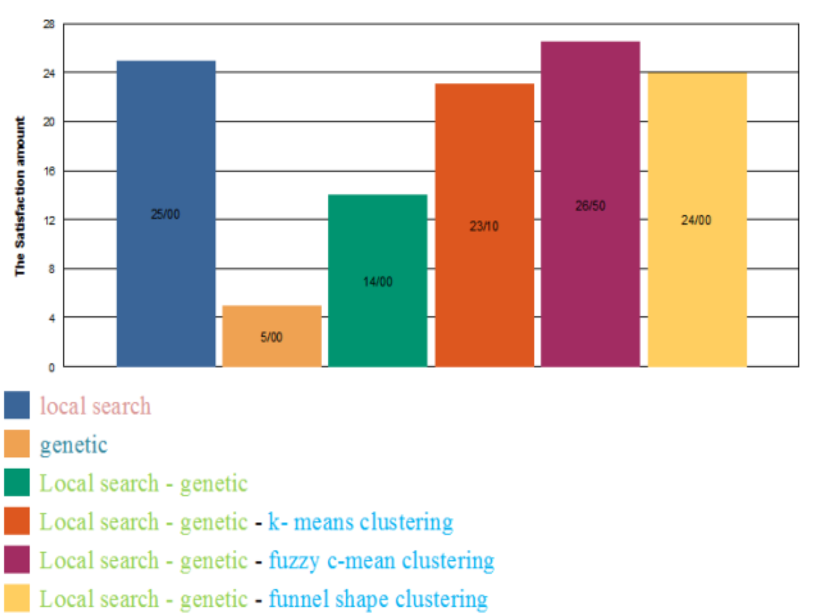


Fig.6. The result of applying local searc algorithm with clustering tecniques

In Fig.7, local search and genetic algorithms with clustering techniques are shown



.

Fig.7. The result of applying local search and genetic algorithms with clustering techniques

In Fig.8, we have shown the genetic algorithm with clustering techniques based on datasets.

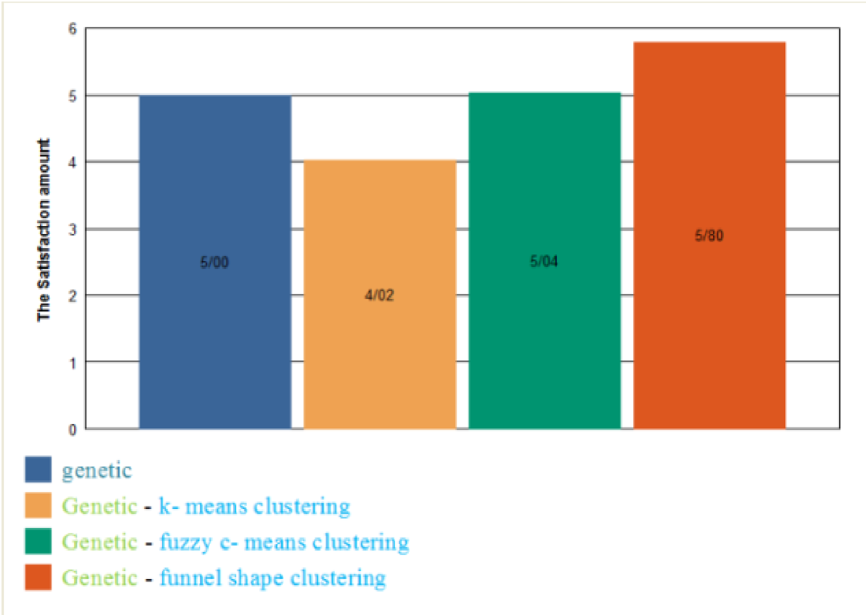


Fig.8. The result of applying genetic algorithm with clustering techniques

Table 1: Comparison of lecturers' constraints satisfaction percent by algorithms

Algorithms	The percent of satisfying	Lecturers soft constraints satisfaction percent
FMCDM- K-means clustering		19.03%
FMCDM- Fuzzy c-means clustering		26.10%
FMCDM- Funnel shape clustering		25.75%
FMCDM		28.00%
Local search		25.00%
FMCDM- Local search		48.00%
K-means clustering		28.00%
FMCDM- Local search- K-means clustering		39.06%
FMCDM- Local search- Fuzzy c-means clustering		**48.04%**
FMCDM- Local search- Funnel shape clustering		46.09%
Local search- K-means clustering		19.03%
Local search- Fuzzy c-means clustering		26.10%
Local search- Funnel shape clustering		25.75%
Genetic		5.00%
Local search- Genetic		14.00%
Local search- Genetic- K-means clustering		23.10%
Local search- Genetic- Fuzzy c-means clustering		26.50%
Local search- Genetic- Funnel shape clustering		24.00%
Genetic- k-means clustering		**4.02%**
Genetic- Fuzzy c-means clustering		5.04%
Genetic- Funnel shape clustering		5.80%
FMCDM : Fuzzy Multi Criteria Decision Making		**Max=48.04%** **Min=4.02%**

5. Conclusion and Future Works

The results obtained from the goals of research by the proposed approach are presented as follows: 1- the proposed method has resulted in descending satisfaction (non-ascending) of lecturers' preferences (soft constraints) for allocation in the additional resources, 2- improved the configuration of soft constraints' weights by using fuzzy values and 3- increased the lecturers' satisfaction by applying the local search method. According to table 1, FMCDM-Local search- fuzzy c-means clustering algorithm and Genetic- K-means clustering algorithm represent the best performance in lecturers' satisfaction with 48.04% and the worst performance in lecturers' violation with 4.02%, respectively. Future works include the following:

- ❖ For two agents of TA and MA, metaheuristic algorithms could be used to increase efficiency in generating and improving timetables.
- ❖ Other clustering algorithms or a combination of these algorithms could be used in two agents CA and TraA to evaluate the current approach with the obtained results in order to find a particular pattern for clustering process.

References

[1] Lewis R. M. R.,Metaheuristics for University Course Timetabling. Ph.D. Thesis, Napier University, Napier, UK, (2006).

[2] Obit J. H., Developing Novel Meta-heuristic, Hyper-heuristic and Cooperative Search for Course Timetabling Problems. Ph.D. Thesis, School of Computer Science University of Nottingham, Nottingham, UK, (2010).

[3] Redl T., A., A Study of University Timetabling that Blends Graph Coloring with the Satisfaction of Various Essential and Preferential Conditions. Ph.D. Thesis, Rice University, Houston, Texas, (2004).

[4] Babaei, H., Hadidi, A., A Review of Distributed Multi-Agent Systems Approach to Solve University Course Timetabling Problem. ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No. 11, ISSN: 2322-5157, pp. 19-28, (2014).

[5] Feizi-Derakhshi, M. R., Babaei, H., Heidarzadeh, J., A Survey of Approaches for University Course TimeTabling Problem. Proceedings of 8th International Symposium on Intelligent and Manufacturing Systems, Sakarya University Department of Industrial Engineering, Adrasan, Antalya, Turkey, pp. 307-321, (2012).

[6] Babaei, H., Karimpour, J., Hadidi, A., A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, pp. 43–59, (2015).

[7] Hafizah A R., Zaidah I., Bipartite Graph Edge Coloring Approach to Course Timetabling. 2010 International Conference on Information Retrieval & Knowledge Management, (CAMP), *IEEE*, pp. 229-234, (2010).

[8] Mayer A., Nothegger C., Chwatal A., Raidl. G., Solving the Post Enrolment Course Timetabling Problem by Ant Colony Optimization. In proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, (2008).

[9] Shahvali Kohshori M., SanieeAbadeh M., Hybrid Genetic Algorithms for University Course Timetabling. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, (2012).

[10] Bykov Y., Petrovic S., A Step Counting Hill Climbing Algorithm applied to University Examination Timetabling. J Sched, (2016).

[11] Rudová H., Müller T., Murray K., Complex University course timetabling. J Sched, 14: pp. 187–207, (2010).

[12] Müller T., Murray K., Comprehensive approaches to student sectioning. Ann Oper Res 181: 249–269, (2010).

[13] Ceschia S., Gaspero L. D., Schaerf A., Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Computers & Operations Research 39: pp. 1615–1624, (2012)

[14] Bellio R, Gaspero L D, Schaerf A (2011) Design and statistical analysis of a hybrid local search algorithm for course timetabling. J Sched 15: 49–61

[15] Müller T., Rudova H., Bartak R., Minimal Perturbation Problem in Course Timetabling. Springer-Verlag Berlin Heidelberg, 3616: pp. 126–146,(2005).

[16] Ceschia S., Gaspero L. D., Schaerf A., The generalized balanced academic curriculum problem with heterogeneous classes. Ann Oper Res, 218:147–163, (2014).

[17] Schaerf A., A survey of automated timetabling. Articifial Intelligence Review, 13 (2):pp. 87–127, (1999).

[18] Bellio R., Ceschia S., Gaspero L. D., Schaerf A., Urli T Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. Computers & Operations Research, 65: 83–92, (2016).

[19] Müller T., University Course Timetabling Solver Evolution. Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016) – Udine Italy, (2016).

[20] Babaei, H., Karimpour, J., Hadidi, A., Applying Hybrid Fuzzy Multi-Criteria Decision-Making Approach to Find the Best Ranking for the Soft Constraint Weights of Lecturers in UCTP. Int. J. Fuzzy Syst.Taiwan Fuzzy Systems Association and Springer -Verlag Berlin Heidelberg, (2017), Doi: 10.1007/s40815-017-0296-z.

[21] Ross j. T., Fuzzy Logic with Engineering Applications. John Wiley & Sons Ltd, University of New Mexico, New Mexico, (2004).

[22] Babaei H., Karimpour J., Mavizi S., Using k-means clustering algorithm for common lecturers timetabling among departments. ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 1, No.19, ISSN: 2322-5157, pp. 86-102, (2016).

[23] Babaei, H., Karimpour, J., Oroji, H., Using fuzzy *c*-means clustering algorithm for common lecturers timetabling among departments. 6th International Conference on Computer and Knowledge Engineering (ICCKE 2016), 978-1-5090-3586 IEEE, October 20-21, Ferdowsi University of Mashhad, (2016).

[24] Babaei H., Karimpour J., Hadidi A., Common lecturers timetabling among departments based on funnel-shape clustering algorithm. Springer Science + Business Media New York, *Applied Intelligence*, DOI 10.1007/s10489-016-0828-5, 46: pp. 386-408, (2017).