ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 1, No.19 , January 2016
ISSN : 2322-5157
www.ACSIJ.org

Using k-means clustering algorithm for common lecturers timetabling among departments

Hamed Babaei[1], Jaber Karimpour [2] and Sajjad Mavizi [3]

[1] Department of Computer Engineering
Islamic Azad University, Ahar Branch
Ahar, Iran
hamedbabaei63@gmail.com

[2] Department of Computer Sciences
University of Tabriz
Tabriz, Iran
karimpour@tabrizu.ac.ir

[3] Department of Computer Engineering
Islamic Azad University, Shabestar Branch
Shabestar, Iran
sajjad.mavizy@gmail.com

Abstract

University course timetabling problem is one of the hard problems and it must be done for each term frequently which is an exhausting and time consuming task. The main technique in the presented approach is focused on developing and making the process of timetabling common lecturers among different departments of a university scalable. The aim of this paper is to improve the satisfaction of common lecturers among departments and then minimize the loss of resources within departments. The applied method is to use a collaborative search approach. In this method, at first all departments perform their scheduling process locally; then two clustering and traversing agents are used where the former is to cluster common lecturers among departments and the latter is to find unused resources among departments. After performing the clustering and traversing processes, the mapping operation in done based on principles of common lecturers constraint in redundant resources in order to gain the objectives of the problem. The problem's evaluation metric is evaluated via using clustering algorithm k-means on common lecturer constraints within a multi agent system. An applied dataset is based on meeting the requirements of scheduling in real world among various departments of Islamic Azad University, Ahar Branch and the success of results would be in respect of satisfying uniform distribution and allocation of common lecturers on redundant resources among different departments.

Keywords: *University Course TimeTabling Problem (UCTTP), Common Lecturer TimeTabling Problem (CLTTP), Multi-Agent Systems, K-mean Clustering Algorithms.*

1. Introduction

The goal of the university course timetabling problem ([1]UCTTP) is to find a method to allocate whole events to fix predefined timeslots and rooms, where all constraints within the problem must be satisfied. Events include students, teachers and courses where resources encompasses the facilities and equipment's of classrooms such as theoretical and practical rooms. Also timeslots include two main components, namely daily and weekly timeslots which it varies from one institution to another. However, each classroom also has its own components allocated to those classrooms (the capacity of theory and practical rooms), number of blackboards and whiteboards related to each theory and practice classroom and etc. [1, 2, and 3].

1.1 Description of the Problem

UCTTP is a hybrid optimization problem in the class of NP-hard problems occur at the beginning of each semester of universities and includes the allocation of events (courses, teachers and students) to a number of fixed timeslots and rooms. This problem must satisfy 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 soft constraints to increase and promote the quality of possible generated timetables as necessary. There are some problems and complexities in UCTTP process; firstly, the scheduling process is an NP-complete problem, then it could not be solved in the polynomial time classes because of the exponential growth of this problem and the existence of some variations in the fast growth of students' numbers in this problem, so we must seek heuristic approaches. Secondly, the number of constraints (hard and soft) in this problem differs from one institution to another. Therefore, the main aim of all of the mentioned algorithms is to maximize the number of soft constraints satisfied in the final timetables [1, 2, 3, and 4].

1.2 The basic definitions of the problem

- Event: a scheduled activity, like: teacher, course, and student.
- Timeslot: a time interval in which each event is scheduled, like: weekly timeslot such as Tuesday and daily timeslot such as 8 a.m. to 9 a.m. and etc.
- Resource: resources are used by events, like: equipment's, rooms, timeslots and etc.
- Constraint: a constraint is a restriction in scheduling of events, categorized into two types of hard and soft constraints, like the capacity of classrooms, given timeslot and etc.
- People: People include lecturers and students and are a part of events.

---
[1] University Course TimeTabling Problem

- Conflict: the confliction of two events with each other, like: scheduling of more than one teacher for one classroom at the same time.

1.3 Different types of constraints in the problem

Constraints in UCTTP problem are classified into two classes of hard and soft constraints. Hard constraints must be satisfied in the problem completely so that the generated solution would be possible and without conflict; no violation is allowed in these constraints. Soft constraints are related to objective function; objective function is to maximize the number of satisfied soft constraints. Unlike hard constraints, soft constraints are not necessarily required to satisfy; but as the number of these satisfied constraints increases, the quality of solutions of objective function increases. In the following, a list of hard and soft constraints presented which are taken from literature [1, 2, 3, 4, 5, 6, and 7].

1.3.1 Hard Constraints

- A teacher could not attend two classes at the same time.
- A course could not be taught in two different classes at the same time.
- A teacher teaches only one course in one room at each timeslot.
- At each daily timeslot in one room only one group of students and one teacher could attend.
- A teacher teaches for only one group of students at each daily timeslot.
- There are some predefined courses which are scheduled in a given timeslots.
- The capacity of the classrooms should be proportional to the number of students of the given course.

1.3.2 Soft Constraints

- The teacher can have the choice to suggest priority certain timeslots for her/his courses either public or private times.
- A teacher may request a special classroom for a given course.
- The courses should be scheduled in a way that the empty timeslots of both teacher and student to be minimized.
- Timetabling of the courses should be conducted in a way that the courses not scheduled at evening timeslots, as it is possible; unless an evening timeslot has been requested by a particular teacher.
- The lunch break is either 12 p.m. to 13 p.m. or 13 p.m. to 14 p.m., usually.
- The start time of classes may be 8 a.m. and the ending time may be 20:30 p.m. (evening), usually.
- The maximum teaching hours for teachers in a classroom are 4 hours.
- The maximum learning hours for students is 4 hours.
- Scheduling should be conducted in a way that one or a group of students not attend university for one timeslot in a day.

1.4 Mathematical formulation of the problem

Formal definition of UCTTP problem includes n: the number of events $E=\{e_1, e_2, \ldots, e_n\}$, k: the number of timeslots $T=\{t_1, t_2, \ldots, t_k\}$, m: the number of rooms $R=\{r_1, r_2, \ldots, r_m\}$, L: the number of rooms' features $F=\{f_1, f_2, \ldots, f_l\}$ and s: the set of students $S=\{s_1, s_2, \ldots, s_s\}$. For example, if the number of daily timeslots is 9 and the number of weekly timeslots is 5, then the total timeslots will be $T= 9 \times 5 =45$.
The input data for each sample problem (data sets) include the size and features of each room, the number of students in an event and information about conflicting events. So, we should know the procedure of measuring violation and non-violation of hard and soft constraints in order to have the ability to replace events within matrixes. At first the penalty function per violation from soft constraint must be calculated for each solution which is corresponding to a timetable, as bellow [3, 5, 6, and 7]:

$$PF\ (S) = \sum_{j=1}^{SC} W_j \times (-1) \qquad (1)$$

In Eq. (1), S is the solution, W_j is the weight of each soft constraint (value 0 means non-violation, value 1 means violation and -1 shows the cost of each violation per soft constraint) and SC is the number of soft constraints. However, PF represents the penalty function. Value of objective function per solution considering hard constraints can be calculated as:

$$OF\ (S) = \sum_{i=1}^{HC} W_i \times (-1) + PF\ (S) \qquad (2)$$

In Eq. (2), W_i is the weight of each hard constraint where value 0 means non-violation, value 1 means violation and -1 shows the cost of each violation per hard constraint. Also HC and OF are the number of hard constraints, and the objective function, respectively. Always the value of first term of right hand side of the Eq. (2) is equal to zero ($\sum_{i=1}^{HC} W_i \times (-1) = 0$), this means that the violation of hard constraints is not feasible. So OF (S) = 0 + PF (S), consequently OF (S) = PF (S).
In order to determine the violation of solutions, from hard and soft constraints, results of sample problems are stored in 5 matrixes namely STUDENT-EVENT, EVENT-CONFLICT, ROOM-FEATURES, EVENT-FEATURES and EVENT-ROOM which is introduced in the following.
Each event is met by each student which is stored in the matrix STUDENT-EVENT. This matrix called matrix A is a $k \times n$ matrix. If the value of $U_{i,j}$ in the matrix $A_{k,n}$ be 1, then student $i \epsilon S$ must attend event $j \epsilon s$, otherwise, its value will be 0. The matrix size is $k \times n = |S| \times n$. The EVENT-CONFLICT matrix is an $n \times n$ matrix with two arbitrary events which could be scheduled in the same timeslots. This matrix called matrix B is used to quickly identify events which potentially allocated to same timeslots. ROOM-FEATURES matrix is a $m \times l$ matrix which shows the features of each room; this matrix called matrix C. If the value of $C_{i,j}$ be 1, then each $i \epsilon R$ has a feature of $j \epsilon F$, and otherwise its value will be 0. The matrix size is $m \times l = m \times |F|$. The EVENT - FEATURE matrix also called matrix D is a $n \times l$ matrix and represents the features required by each event. Namely, event $i \epsilon E$ requires features of $j \epsilon F$, if and only if $d_{i,j}$=1. The matrix size is $n \times l = n \times |F|$. Finally the EVENT-ROOM matrix called G

matrix is an n × m matrix which represents the list of possible rooms so that each event could be allocated in those rooms. This matrix represents the quick identification of all rooms in terms of their size and features for each appropriate event. The matrix size is n × m [1, 3, 5, 6, and 7].

1.5 The approaches used in the study of UCTTP

The first definition of timetabling has been presented as three sets of: 1) teachers, 2) classrooms and 3) timeslots (Gotlib, 1963). Approaches used to solving the UCTTP problem up to now are as follows: 1) Operational Researches (OR) based techniques including graph coloring theory based technique, IP/LP method and Constraint Based Satisfaction(s) technique (CPSs); 2) Metaheuristic approaches also including Case Base Reasoning method (CBR), population based approaches and single solution based approaches where the population based approaches includes Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Memetic Algorithm (MA), Harmonic Search Algorithm (HAS) and single solution algorithms also includes Tabu Search Algorithm (TS), Variable Neighborhood Search (VNS), Randomized Iterative Improvement with Composite Neighboring algorithm (RIICN), Simulated Annealing (SA) and Great Deluge Algorithm (GD); 3) multi criteria and multi objective approaches; 4) intelligent novel approaches such as hybrid approaches, artificial intelligence based approaches, fuzzy theory based approaches and 5) distributed multi agent systems approach [1, 2, 3, 5, 6, and 7].

1.6 Motivation and historical perspective of the problem

Agents are technologies inspired from global environment to develop initial instances of systems. Whenever a distributed multi agent system is considered, it means that there is a network of agents cooperates with each other to solve problems which are out of capability of each single agent [8]. Recently, using distributed multi agent systems based approach to solve UCTTP problem has been applied by [9] where in the this method, a solution is used to deal with UCTTP problem using distributed environment and an interface agent -which is responsible to cooperate different timetabling agents- collaborate with each other to improve the solution of common goal. The initial timetables are generated for multi agent systems by using multiple hybrid metaheuristics which are a combination of graph coloring metaheuristics and local search in different methods. The hybrid metaheuristics provide the capability to generate possible solutions for all samples of both Socha et al. (2002) and international competitions timetabling 2002 datasets. However, recently, [10] has used distributed agents to create UCTTP by considering hard (necessary) and soft (desirable) constraints. Also, he presented fairly meeting of distribution in allocating resources in his Ph.D. thesis. There are two types of agents in that model which are year- programmer agent and rooms' agent. However, there are four principles to efficiently organize agents, including: 1) queue and the sequential queue algorithm, 2) queue and interleaved queue algorithm, 3) round robin and sequential round robin algorithm and 4) round robin and interleaved round robin algorithm. The problem formulation and dataset have been adopted from the third section of ICT-2007. The obtained result ensures the consistency of interleaved round robin principle for year-programmer agents in the system and the fairest chance in obtaining the required resources. However, recently [11] has used a novel clustering technique based on FP-Tree to solve UCTTP where the given technique is done to classify students based on their selective courses who submitted for the next semester. The aim of this clustering is to solve scheduling of courses where in the previous semesters the submission of students in some courses due to simultaneous scheduling has been prevented, while in this technique no conflict would happen over scheduling of exams since no two exams at the same time would be taken for courses by two identical groups of students.

1.7 Claim

In this article our main goal is to schedule common lecturers ([۲]CLTTP) among different departments based on redundant resources among departments. Clustering algorithms have been used to schedule common lecturers within a distributed system based approach. Since the system uses a distributed multi agent architecture so in order to reach the goal of CLTTP problem, two agents, clustering and traverser, are considered, respectively. The clustering agent performs the act of clustering common lecturers among departments within clusters according to the common, semi-common and uncommon priorities, constraints and features of lecturers so that lecturers who are similar and closer to each other in terms of selecting priorities and constraints are places within high value clusters (primary and more dense clusters) in order to be allocated to their demanded and prioritized resources. After clustering process, the mapping of these clusters is done due to the clusters of common lecturers among departments in to traversed groups of redundant resources among departments collected by traverser agent. The research performed in this article is to present a new and different approach of timetabling problem to develop and make the process of timetabling common lecturers among departments over existing (redundant) resources in departments of a university scalable. The contributions presented in this article to solve the CLTTP problem include: 1) descending satisfaction (from desirable to undesirable priorities) of constraints and priorities of common lecturers among departments and 2) minimizing the loss of redundant resources among departments. Of course, these goals are evaluated by using clustering common lecturers among departments and grouping the redundant resources among departments.

2. Related works

Those approaches solved UCTTP problem by now include the mentioned methods in section 1.5.

2.1 Operational research approaches

Graph coloring approach is on how to model a UCTTP problem by using a non-directional graph where [12] has used vertices as events, colors as time slots and edges as constraints in a graph to solve timetabling problem where no two adjacent vertices have co-colors; since a sign of conflict has been authenticated in the time table. Another hybrid approach has also been proposed to solve UCTTP problem using genetic algorithm by [13] which reduces the cost of finding the number of minimum required colors to color a graph with this hybrid method. In [14], IP method (integer programming) has

[۲] Common Lecturer TimeTabling Problem

been presented to solve UCTTP problem where the goal is to allocate a set of courses among lecturers and groups of students and also a set of weekly and daily time period pairs. Again, [15] has presented an IP-based two-step simplification method where during step 1, the classes require sequence are scheduled by allocating courses to given days and times and during step 2, ensuring the sequence of those courses requiring more than one time period for the same student groups is also done.

2.2 Meta heuristic approaches

In [16], a genetic algorithm has been used in respect of ordering a university timetabling where the intersection rate was 70% and no hard constraint was violated and the applied constraints were almost on room's occupation and capacity. However, [17] has proposed a new GA technique to solve UCTTP problem which uses a learner machine. The results of this technique include minimization of the number of violated soft constraints, high usage of available rooms and reduction of lecturers' workload. Of course applying ant colony optimization algorithm by [18] to UCTTP problem after submission has been done according to ITC-2007 dataset where ants allocate events to rooms and time slots based on two types of pheromone τ_{ij}^{s} and τ_{jk}^{y} . This algorithm has performed well on timetabling and generated good results during longer. Applying a hybrid ant colony system has been proposed to solve UCTTP problem in [19], where two types of hybrid ant systems including combination of SA with AC and combination of TS with AC have been presented. A number of ants perform entire allocation of courses to time slots based on a predefined list. Selection of time slots' probabilities is done by ants to allocation courses using heuristic information and an indirect coordinator mechanist among agents (Stigmergic) and existing activities within an environment. The memetic algorithm has been done using [20] to solve UCTTP problem via combination of local search method in genetic algorithm. One of the local searches is done on events and the other one is performed on time slots.

The Tabu search algorithm has been applied by [21] for the first time to allocate students to courses and also balance the number of students within whole submitted group where the first phase is: generating a set of solutions for a student, and the second phase is: combining a set of solutions and applying Tabu search with local strategies and the third phase is also: allocating room and improving allocation, while without changing the initial allocation of courses to timeslots. In [22], the influence of neighborhood structures has been presented on Tabu search algorithm to solve UCTTP problem where the effect of simple and swap transitions has been tested on Tabu search operations based on neighborhood structures. Here, four new neighborhood structures have been used and compared. To solve UCTTP problem, the combination of kempe neighborhood chain has been presented in simulated annealing algorithm by [23] where one of the hard constraints of reformulation is done by relaxation and then this constraint is created in the form of relaxed soft constraint. However, the relaxation problem is analyzed in two steps: 1- to create a feasible solution, a heuristic based graph is used and 2- a simulated annealing algorithm has been used to minimize the violations of soft constraints (in the second phase, a kempe neighborhood chain based heuristic has been used).

[24] Also has used directed local search strategy in genetic algorithm to solve UCTTP problem where the directed search strategy uses a data structure to create offspring that stores the extracted information of good individuals of previous generations in itself. The results are satisfactory with this local search combined in the genetic algorithm. The aim is to maximize allocations and minimize the violations from soft constraints. The variable neighborhood search algorithm (VNS) has been presented by [25] to solve UCTTP problem which proposes the base VNS and then states some modifications to each solution which apply an exponential Monte Carlo acceptance criterion. However, the main idea of applying Monte Carlo acceptance criterion was to improve the heuristics by admitting the best solution with given probability so that the number of promised neighbors would be found.

2.3 Modern intelligent approaches

A hybrid algorithm has been presented by [26] which is the combination of sequential heuristic and simulated annealing to solve UCTTP problem on ITC-2002 dataset. This method includes three phases: Phase 1: using a sequential heuristic to generate feasible time tables; phase 2: applying simulated annealing to minimize the number of soft constraints' violations and phase 3: uses simulated annealing to increase the improvement of the generated time tables' quality. Recently, a multi population hybrid genetic algorithm has been proposed by [27] to solve UCTTP problem based on three genetic algorithms FGARI, FGASA and FGATS. In this algorithm, fuzzy logic is used to evaluate the number of violations from soft constraints in fitness function to deal with real worlds data which are ambiguous and non-deterministic and random methods, local search, simulated annealing and Tabu search would also be beneficial in addition to fuzzy method to improve inductive search in order to meet the need of search ability.

To solve UCTTP problem, [28] has presented a fuzzy multi criteria heuristic ordering method where the ordering of events has been done according to three independent heuristics simultaneously using fuzzy methods. The sequential combination of three heuristics is ordered as follows: 1- the highest degree, 2- saturation degree and 3- enrollments degree and the fuzzy weight of an event is also used to represent what problem the event has to be scheduled. The ordered events are allocated to the last time slot with the least value of penalty cost as a descending manner while the feasibility is maintained throughout whole process. A fuzzy solution has been presented by [29] based on memetic approach to solve university timetabling where a time table has been compared with both genetic and memetic algorithm and its results may satisfy the existing constraints simultaneously in a shorter time interval. The aim was to use fuzzy logic as a tool to local search in memetic algorithm. [30] Has proposed the fuzzy genetic heuristic idea to solve UCTTP problem where the genetic algorithm has been applied by using indirect representation based on the features of integrating events and modeling the fuzzy set to evaluate the violation from soft constraints in the objective function according to uncertainty of real world data. Here, a degree of uncertainty which is in an objective function is considered for each soft constraint and this uncertainty is evaluated by formulation of soft constraint violation parameter in objective function by using fuzzy membership functions.

ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 1, No.19 , January 2016
ISSN : 2322-5157
www.ACSIJ.org

3. The proposed method

In [8], an agent could observe and receive everything through sensors from its environment and then perform within environment via the stimulus. Agents are classified into various classes based on their applications including the following agents: 1- autonomous, 2- intelligent, 3- reactive, 4- pro-active, 5- learner, 6- mobile, 7- collaborative/communicative. So, agents must have a common language and a communicative media to communicate and cooperate with each other where these two components are vital among agents.

3.1 Common lecturers timetabling problem among departments

Common lecturers' timetabling problem among departments is one of the challenges among university departments where in this article it has been tried to perform this scheduling based on regarding priorities and requirements of common lecturers to allocate redundant resources among departments. Since the common lecturers among departments always deal with facilitating their timetabling, then a new idea and solution must be researched to facilitate the timetabling of common lecturers so that some challenges such as collision among lecturers and other common events in departments and not promoting the satisfaction of common lecturers based on their desirable choices would be avoided. However, solving CLTTP problem has led to a developed and scalable scheduling process where in this research we have considered this by performing scheduling and distribution of common lecturers over redundant resources among departments. Therefore, to solve a CLTTP problem, the solution in the form of distributed multi agent system accompanied with applying clustering algorithms must follow the process of minimizing the collision of common lecturers among departments. Fig. 1 represents a holistic view on CLTTP problem in a tree structure.

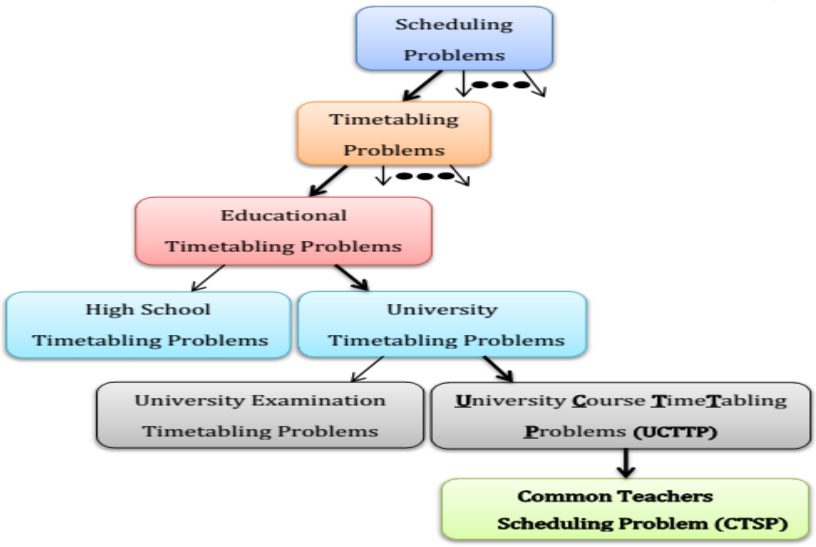


Fig. 1: The tree structure of common lecturers' timetabling problem among departments

3.2 Frameworks and infrastructures of the proposed algorithm

The proposed algorithm consists of four agents: 1- time table (each ith department or agent, $TA_i; i = 1,2,\ldots$), 2- mediator agent (MA), 3- clustering agent (CA) and 4- traverser agent (TraA) which have been shown in Fig. 2, with their relations in three phases. The first phase includes steps 1 and 2 which are planned by the timetabling agent to produce feasible with no conflict time tables. Of course, in this phase, the identification and collection of common lecturers among departments is done by the mediator agent in step 3, the second phase includes steps 4, 5 and 6 which performs the process of clustering common lecturers among departments within the clustering agent to make uniform distribution on the traversed redundant resources of each department by the traverser agent and the third phase consists of steps 7 and 8 where the process of mapping the common lecturers' clusters is done in redundant resources based on the constraints of common lecturers and send the time tables with the capability of planning to each department for a semester.

3.2.1 The first phase

The first phase includes the hard constraints related to lecturers of each department satisfied by TA_i agent and contains the following constraints: 1- a lecturer could not teach more than 6 hours per day, 2- a lecturer could not be in more than one department at the same time slot, simultaneously, 3- a lecturer could not be in two classes at one or more departments in one day or at the same time slot, 4- a class is allocated to one lecturer at one time slot, and 5- two lecturers could not be in the same class of a department at the same time. Fig. 3 show the lecturers timetabling algorithms on the resources related to each department by TA_i agent. Between the first and the second phases, the mediator agent (MA) studies the operation of extracting common lecturers among departments accompanied with their features to cluster in the next step without any conflict based on the aim of the problem which is to time table the common lecturers among departments and sends them to their related departments (TA_i) in order to modify the conflicts when it discovers a conflict and inconsistency in the time tables of common lecturers among departments. And then the time tables of common lecturers of each department fixed in the respect of the problem aim by the mediator agent are sent during step 3 to the clustering agent (CA).

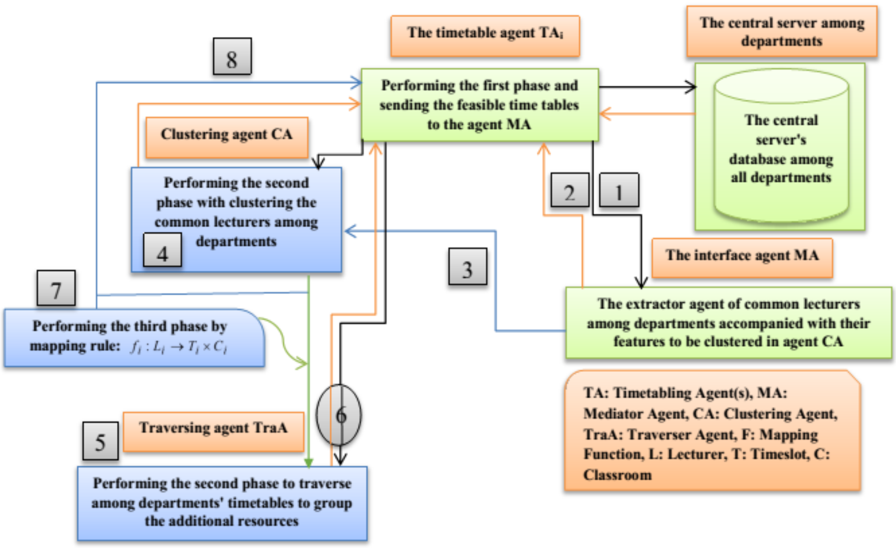


Fig. 2: The general view of CLTTP problem's schematic

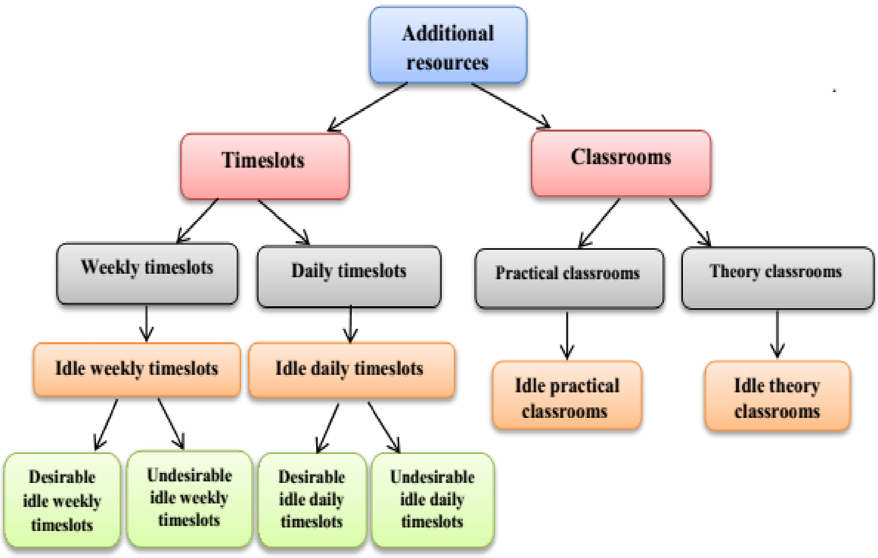


Fig. 3: The structure of grouping redundant resources among departments in TraA

3.3 The second phase

In the second phase, CA clusters common lecturers among departments based on their constraints (step 4) and TraA agent is applied through traversing and grouping the redundant resources among departments (among TA_i agents) (step 5). Of course, before entering step 5, all busy and redundant resources have been determined entirely through time tables of each department (TA_i) in step 6 and sent to step 5 by TraA agent to perform traversing and grouping. In the second phase, two ideas have been proposed where the former is to consider two new agents of CA and TraA in the architecture of multi agent system and perform the mapping process by CA in TraA and the latter is to state a clustering method coinciding the type of problem called k-means clustering to perform the process of clustering common lecturers among departments applied within CA agent. The algorithms of two CA and TraA agents have been shown in figures 4 and 5. The third phase

In the third phase, the process of mapping priorities and requirements of common lecturers is presented to uniformly distribute and allocate redundant resources among departments. In the last step of the third phase (step 8) the final solution (timetabling of common lecturers among departments for one semester) is sent to all the departments based on each department's (TAi) identification codes after the process of mapping clustering agents in the traversed redundant resources in TraA agent.

3.3.1 Clustering and traversing in the second phase

In the second phase, the clustering of common lecturers among departments is performed in the clustering agent (CA) by two algorithms of k-means, fuzzy c- means clustering and the proposed funnel-shape clustering where the clustering process is described through four features of each common lecturer as: desirable daily time slots, undesirable daily time slots, desirable weekly time slots and undesirable weekly time slots. Following the process of clustering common lecturers among departments based on their features, traversing and grouping of the redundant resources is done by TraA agent as Fig. 3.

3.4 The complete description of adapted k-means clustering algorithm's details

After stating the priorities and soft constraints of each common lecturer among departments based on (3), now in (3) let consider L_k as the kth common lecturer, $WeeklyTimeslots_k$ as the kth weekly time slot, $DailyTimeslots_k$ as the kth daily time slot, $Departement_k$ as kth department and χ_k as the membership degree of each common lecturer.

$$L_k = \left\{ {WeeklyTimeslot_k}\big/{Departement_k} = \left\{ {\chi_k}\big/{DailyTimeslot_k} \right\} \right\} ; \chi_k \in [0,1] \tag{3}$$

In (4), the default pattern of primary matrixes is represented as $U^{(0)}_{Dep/WeeklyTimeslots}$ related to each department and each weekly timeslot and the values of membership degree of each common lecturers is denoted by χ_{ik} per row or daily timeslot per department are represented as following: each daily time slot $DailyTimeslots_{(1-7)}$ from 8-9:30$_1$ to 19-20:30$_7$ as one cluster which would be 7 clusters and weekly timeslots $WeeklyTimeslots_{(1-7)}$ from Saturday (1) to Friday (7) and Dep as five departments $Departement_{(1-5)}$ in (4). Finally we would reach to the final matrix of $U^{(0)}$ consisting of 7 rows (clusters) and 30 columns (common lecturers). The resulted matrix is represented as (5). In k-means clustering, the membership degree and no membership of each kth common lecturer in the ith cluster would be computable based on (5).

$$U^{(0)}_{Dep_{(1-5)}/WeeklyTimeslots_{(1-7)}} = \left\{ \begin{array}{l} 8-9:30_1 \\ 10-11:30_2 \\ 12-13_3 \\ 13-14:30_4 \\ 15-16:30_5 \\ 17-18:30_6 \\ 19-20:30_7 \end{array} \right. \begin{bmatrix} \chi_{11} & \cdots & \chi_{1,30} \\ \cdots & & \cdots \\ & \cdots & \\ \cdots & & \cdots \\ & \cdots & \\ \cdots & & \cdots \\ \chi_{71} & \cdots & \chi_{7,30} \end{bmatrix}_{7,30} \tag{4}$$

$$\chi_{A_i}(x_k) = \begin{cases} 1; x_k \in A_i \\ 0; x_k \notin A_i \end{cases} ; A_i \, is \, ith \, cluster \, and \, x_k \, is \, kth \, common \, lecturer \tag{5}$$

In (5), parameter $\chi_{A_i}(x_k)$ meaning the membership degree of x_k (kth common lecturer) in cluster A_i (ith cluster) is represented with two binary values of 0 or 1 to no-membership and membership of kth common lecturer in ith cluster. However, in order to use (5) within the membership matrix, we could review (6). In (6), parameter χ_{ij} is extended as parameter $\chi_{A_i}(x_j)$ where i represents the number of clusters and j is considered as the number of common lecturers. After stating all features and priorities of each common lecturer among departments, we would reach a final matrix of $U^{(0)}$ based on matrixes of (4) in terms of department, daily time slots and weekly time slots parameters which includes 7 rows (clusters) and 30 columns (common lecturers). The obtained matrix is represented as (7).

$$\begin{gathered} \chi_{ij} = \chi_{A_i}(x_j); \\ i = 1, \ldots, c; \; j = 1, \ldots, n, \, and \, x_j \, is \, jth \, common \, lecturers \, in \, ith \, cluster \end{gathered} \tag{6}$$

$$U^{(0)}_{Departements/Weekly\ Timeslots} = \underbrace{\left\{ \begin{array}{r} 8-9:30 \\ 10-11:30 \\ 12-13 \\ 13-14:30 \\ 15-16:30 \\ 17-18:30 \\ 19-20:30 \end{array} \right.}_{\text{Clusters}} \overbrace{\begin{array}{cccccccccccc} L_1 & L_2 & L_3 & & & & & & & L_{28} & L_{29} & L_{30} \end{array}}^{\text{Common Lecturers}} \begin{bmatrix} \chi_{11} & \chi_{12} & \chi_{13} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{1,28} & \chi_{1,29} & \chi_{1,30} \\ \chi_{21} & \chi_{22} & \chi_{23} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{2,28} & \chi_{2,29} & \chi_{2,30} \\ \chi_{31} & \chi_{32} & \chi_{3,3} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{3,28} & \chi_{3,29} & \chi_{3,30} \\ \chi_{41} & \chi_{42} & \chi_{43} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{4,28} & \chi_{4,29} & \chi_{4,30} \\ \chi_{51} & \chi_{52} & \chi_{53} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{5,28} & \chi_{5,29} & \chi_{5,30} \\ \chi_{61} & \chi_{62} & \chi_{63} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{6,28} & \chi_{6,29} & \chi_{6,30} \\ \chi_{71} & \chi_{72} & \chi_{73} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{7,28} & \chi_{7,29} & \chi_{7,30} \end{bmatrix}_{7\times 30} \tag{7}$$

3.4.1 The steps of adapted k-means clustering for CLTTP problem

By given initial matrix of $U^{(0)}$ for each common lecturer among departments, we have the following steps:
Finding the centers of each i cluster according to j feature of each common lecturer among departments would be calculated through (8). In (8). We could find the center of each i cluster per each j feature and the priorities of common lecturers among departments. In (8) parameters $k=1,...,n$, X_{kj} and χ_{ik}, represent the number of common lecturers among departments, the membership degree of each common lecturer due to ith cluster and the contribution amount of each kth common lecturer to jth feature. Equation (9) is the extension of variable X_{kj} of each common lecturer over three parameters or features (priority) X_{kj_1} of departments, X_{kj_2} is the daily time slot and X_{kj_3} is the weekly time slot. After describing the structure of each i feature of common lecturers in (9), now the rule of finding the center of cluster must be presented in terms of (10) which is consistent with the common lecturers' timetabling problem. It must be noted that since the common lecturers have been distributed among departments, then the (8) must be cycled among all five departments ($Dep=1,...,5$) based on three features and priorities of each common lecturer determined by parameter v_i per given common lecturer.

$$V_{ij}=\frac{\sum_{k=1}^{n}\chi_{ik}\times X_{kj}}{\sum_{k=1}^{n}\chi_{ik}} \tag{8}$$

$$\sum_{k=1,\,j=1}^{k=n,\,j=3}X_{kj}=\sum_{k=1}^{n}\left(X_{kj_1}+X_{kj_2}+X_{kj_3}\right) \xrightarrow{j_1,j_2,j_3}\left(j_1 isDepartements,\ j_2 isWeeklyTimeslots,\ j_3 isDailyTimeslots\right) \tag{9}$$

$$V_{ij}=\sum_{Dep=1}^{5}\left\{\frac{\sum_{k=1}^{n}\left(\chi_{ik}\times\left(\sum_{j=1}^{3}X_{kj}\right)\right)}{\sum_{k=1}^{n}\chi_{ik}}\right\};v_i=\left\{v_{i_1},v_{i_2},\ldots,v_{i_j}\right\}i=1,\ldots,7;\ j=1,\ldots,3 \tag{10}$$

Obtaining the distance of each k common lecturer out of cluster i over the lecturer placed in the center of cluster i and extending the distance of k- means clustering would be used to be compatible with common lecturers timetabling problem based on (11). In (11), let d_{ik} be the distance parameter of kth common lecturer over ith cluster and two parameters X_{kj} and V_{ij} represent the ratio of kth common lecturer over each feature j (department, weekly time slot and daily time slot), respectively and the other parameter would be the variable of finding the center of ith cluster over feature j of each kth common lecturer.
Now, we must obtain the updating process of initial matrix's elements $U^{(0)}$ called the values of membership degree in order to reach matrix $U^{(1)}$. Equation (12) would be the main rule to update the elements of matrix $U^{(0)}$ namely χ_{ik} s in k- means clustering.

$$d_{ik}=d(X_k-V_i)=\sqrt{\sum_{j=1}^{3}\left(X_{kj}-V_{ij}\right)^2}\xrightarrow{k=1,\ldots,30;i=1,\ldots,7} \sqrt{(X_{k1}-V_{i1})^2+(X_{k2}-V_{i2})^2+(X_{k3}-V_{i3})^2} \tag{11}$$

$$Update: U^{(r)};\forall i,k:\chi_{ik}^{(r+1)}=\begin{cases}1; d_{ik}^{(r)}=\min\left\{d_{ik}^{(r)}\right\}\forall i\in c,\\ k=1,\ldots,n; i=1,\ldots,c; r=0,1,\ldots\\ 0; theOtherwise\end{cases} \tag{12}$$

Equation (12) performs the updating process of the initial matrix's elements $U^{(0)}$. The parameters of (12) are as follows: r is the counter and repeater of updating, i as ith cluster ($i=1,...,c$), k means the kth common lecturer ($k=1,...,n$), $d_{ik}^{(r)}$ as rth iteration of the rule of finding the distance of kth common lecturer over ith cluster and $\chi_{ik}^{(r+1)}$ represents the membership degree of kth common lecturer over ith cluster after the first iteration and upon the initial matrix $U^{(0)}$ (described either randomly or based on the requirements of each common lecturer). In (13), the process of extending the rule of updating the (12) is presented based on the common lecturer timetabling problem. In (12), that element of initial matrix $U^{(0)}$ with the minimum value in terms of distance of kth lecturer over ith cluster, $\min_{i=1,\ldots,c;k=1,\ldots,n;r=0,1,\ldots}\left\{d_{ik}^{(r)}\right\}$, equals to $\chi_{ik}=1$ and otherwise it would be $\chi_{ik}=0$. In (12) the value of χ_{ik} would be 0 or 1. In fact, according to (13), $d_{ik}^{(r)}$ with the minimum value is replaced with 1 and others with 0. Now, after computing each updated value of χ_{ik} based on (12), matrix $U^{(1)}$ is formed as (14).

$$\begin{aligned}&k=1 \Rightarrow \min(d_{11},\ldots,d_{71}) = \chi_{11} or \ldots or \chi_{71}\\ &\ldots\ldots\ldots\ldots\\ &k=30 \Rightarrow \min(d_{1,30},\ldots,d_{7,30}) = \chi_{1,30} or \ldots or \chi_{7,30}\end{aligned} \tag{13}$$

Common Lecturers

Clusters

$$U^{(1)}_{\text{Departements}/\text{Weekly Timeslots}} = \begin{array}{c} \\ 8-9{:}30\\ 10-11{:}30\\ 12-13\\ 13-14{:}30\\ 15-16{:}30\\ 17-18{:}30\\ 19-20{:}30 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} L_1 & L_2 & L_3 & & & & & & & L_{28} & L_{29} & L_{30} \end{array}\\ \begin{bmatrix} \chi_{11} & \chi_{12} & \chi_{13} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{1,28} & \chi_{1,29} & \chi_{1,30}\\ \chi_{21} & \chi_{22} & \chi_{23} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{2,28} & \chi_{2,29} & \chi_{2,30}\\ \chi_{31} & \chi_{32} & \chi_{33} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{3,28} & \chi_{3,29} & \chi_{3,30}\\ \chi_{41} & \chi_{42} & \chi_{43} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{4,28} & \chi_{4,29} & \chi_{4,30}\\ \chi_{51} & \chi_{52} & \chi_{53} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{5,28} & \chi_{5,29} & \chi_{5,30}\\ \chi_{61} & \chi_{62} & \chi_{63} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{6,28} & \chi_{6,29} & \chi_{6,30}\\ \chi_{71} & \chi_{72} & \chi_{73} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \chi_{7,28} & \chi_{7,29} & \chi_{7,30} \end{bmatrix}_{7\times 30} \end{array} \tag{14}$$

At each step, in order to terminate the updating process of matrix $U^{(r)}$ to $U^{(r+1)}$, (15), the matrix norm rule, must be used to terminate the execution of k- means clustering algorithm. However, it must be said that the iteration process of (15) is in a way that we would reach to an optimal solution matrix $U^{(*)}$ and this procedure follows the $U^{(r=0)} \to U^{(r=1)} \to \ldots \to U^{(*)}$ rule. Equation (16) represents how to apply (15) for common lecturer timetabling problem. Step 4 is the final phase of k-means algorithm for membership of each common lecturer where the (14) fails, the restoration would continue from the step 1 with recently created matrix $U^{(1)}$ so that we reach a new matrix $U^{(2)}$ which is the updated matrix $U^{(1)}$ and so on.

After terminating each membership matrix U 's updating, the value of objective function must be obtained in terms of two parameters χ_{ik} and d_{ik} based on (17), where k is the number of common lecturers, c is the number of clusters, χ_{ik} is the membership degree of each kth common lecturer in ith cluster and d_{ik} is also the distance of kth common lecturer over the common lecturers within the center of ith cluster in c cluster.

$$\left\| U^{(r+1)} - U^{(r)} \right\| = \max_{i,k} \left| \chi_{ik}^{(1)} - \chi_{ik}^{(0)} \right| \le \varepsilon_L ; \qquad \varepsilon_L = 0.1 or 0.01, r = 0,1,2,\ldots \tag{15}$$

$$J(U,v) = \sum_{k=1}^{n=30} \sum_{i=1}^{c=7} \chi_{ik} \left(d_{ik}^2 \right) Objective function \tag{16}$$

$$\left\| U^{(r=1)} - U^{(r=0)} \right\| = \max_{i=1,\ldots,7; k=1,\ldots,30} \begin{vmatrix} \chi_{11}^{(1)} - \chi_{11}^{(0)} & \cdots & \cdots & \chi_{1,30}^{(1)} - \chi_{1,30}^{(0)}\\ \chi_{21}^{(1)} - \chi_{21}^{(0)} & \cdots & \cdots & \chi_{2,30}^{(1)} - \chi_{2,30}^{(0)}\\ \chi_{31}^{(1)} - \chi_{31}^{(0)} & \cdots & \cdots & \chi_{3,30}^{(1)} - \chi_{3,30}^{(0)}\\ \chi_{41}^{(1)} - \chi_{41}^{(0)} & \cdots & \cdots & \chi_{4,30}^{(1)} - \chi_{4,30}^{(0)}\\ \chi_{51}^{(1)} - \chi_{51}^{(0)} & \cdots & \cdots & \chi_{5,30}^{(1)} - \chi_{5,30}^{(0)}\\ \chi_{61}^{(1)} - \chi_{61}^{(0)} & \cdots & \cdots & \chi_{6,30}^{(1)} - \chi_{6,30}^{(0)}\\ \chi_{71}^{(1)} - \chi_{71}^{(0)} & \cdots & \cdots & \chi_{7,30}^{(1)} - \chi_{7,30}^{(0)} \end{vmatrix} \le \varepsilon_L \tag{17}$$

Before mapping these functions, the way of independently mapping of function g has been shown in Fig. 4 for the resources of each department and the function f within Fig. 5 has been represented to map the common lecturers among departments in additional resources.

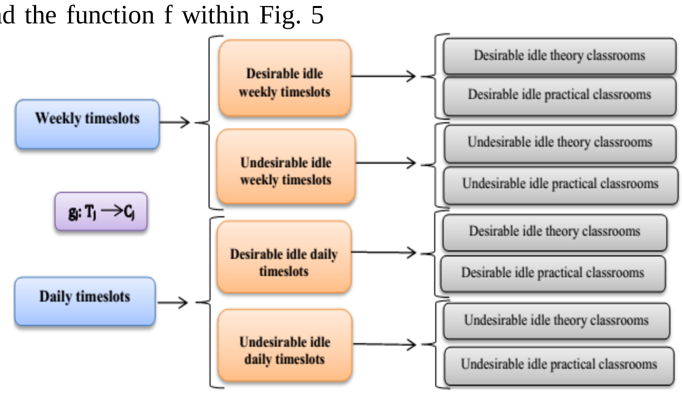


Fig. 4: Mapping time slots in classes

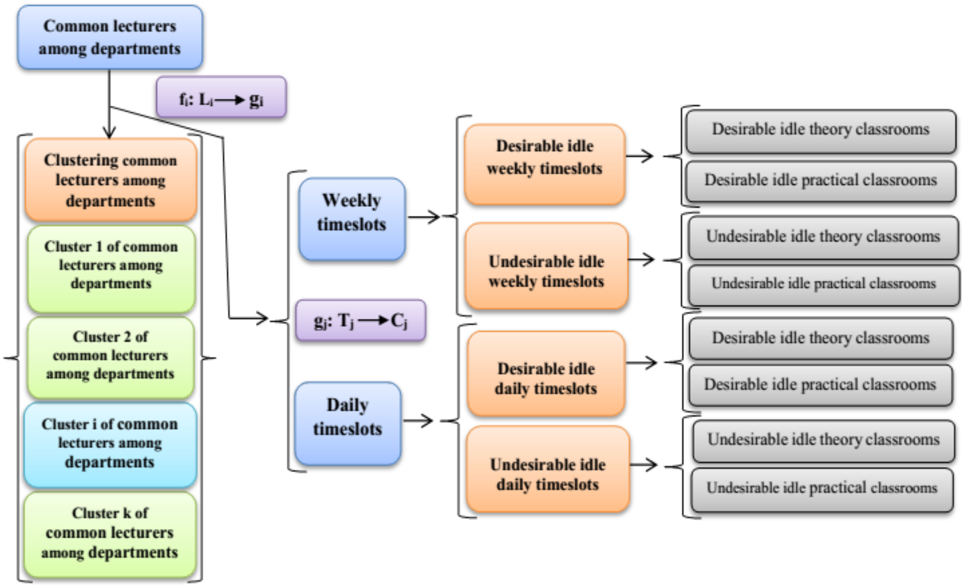


Fig. 5: Mapping the clusters of common lecturers in additional resources among departments

3.5 Mapping

In the third phase the mapping function is as $f_i : L_i \rightarrow T_i \times C_i$, where f_i is the mapping function of priorities and requirements of common lecturers (soft constraints of common lecturers), L_i s are the representative clusters of common lecturers among departments, T_i s represent additional time slots among departments and C_i s also represent additional classes among departments. However, before mapping function f, the mapping of function f must also been performed by agent $TraA$ for the resources among departments as $g_j : T_j \rightarrow C_j$. Fig. 6 presents the way of mapping two functions f and g for the common lecturers to the additional resources among departments.

4. Results and experiments

To test the structure of the proposed algorithm, we consider a data set including 30 lecturers, 5 departments (computer engineering, electronic engineering, civil engineering, humanity science and mathematics), 7 weekly time slots (Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday), 7 daily time slots (8-9:30, 10-11:30, 12-13, 13-14:30, 15-16:30, 17-18:30 and 19-20:30) and 13 classrooms per department (3 practical classes an 10 theoretical classes). The properties of the system to implement include a CPU with 2.13 GHZ speed, 3GB RAM and Win7 operating system and the implementation tools also include 1) C#.net 2010 programming language, 2) using SQL server 2008 software for querying from the databases and 3) reporting by Crystal Report v.13. Total number of resources in the university equals to $7\times7\times5\times(10+3)$ and if we want to calculate the separate resources of each department we would have $(7\times7\times5\times(10+3))\div5$ and the total number of the remained additional resources is obtained as $[(7\times7\times5\times(10+3))-(7\times7\times(10+3))]$. The k-means clustering algorithm must be performed to find the loss percent of additional resources per department so that the minimized percent of additional resources per department, $\frac{Dep_D}{(7\times7\times(10+3))}\times100, D=1,\ldots,5$, is obtained as the dedicated resources of each department divided by whole resources of departments, therefore, each D^{th} department minimizes the loss percent of its additional resources.The criteria of evaluating the CLTTP problem's purposes After using the k-means clustering algorithm and allocating to (additional) resources, the following relations are presented to evaluate the criteria of the paper. Equation (18), $CTDS_1^{(i)}$, computes the descending satisfaction percent of each common lecturer among departments' features at each cluster and (19), $CTDS_2^{(i_j)}$, also obtains the descending satisfaction percent of each common lecturer among departments' priorities and features among clusters and over each cluster.

(18) is calculated per cluster. The numerator of (18) means how many requirements and features of the k^{th} common lecturer in i^{th} cluster presented initially as a report (selections and requirements of each department also could be considered) have been satisfied and the denominator of (18) represents the total number of requests, priorities and requirements of k^{th} common lecturer at that i^{th} cluster which is the sum of satisfied priorities and requirements accompanied with the dissatisfied priorities at i^{th} cluster for the k^{th} common lecturer and the satisfaction percent of k^{th} common lecturer's feature is obtained at i^{th} cluster.

ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 1, No.19 , January 2016
ISSN : 2322-5157
www.ACSIJ.org

$$CTDS_1^{(i)} = \frac{\sum_{k=1}^{n} W_{ik}^{SC} \times X_{ik}}{\sum_{const=1}^{r} \left(Total_{const}^{SC}\right)} \times 100 = \frac{\sum_{Const=1}^{r} \sum_{k=1}^{n} \left(W_{ik}^{Const} \times L_{ik}\right)}{\sum_{Const=1}^{r} \sum_{k=1}^{n} Total_{k}^{Const}} \times 100$$
$$L_{ik} = \sum_{k=1}^{n=30} \sum_{j=1}^{m=3} X_{k_j} = \sum_{k=1}^{n} \left(X_{k_1} + X_{k_2} + X_{k_3}\right) = \left(X_{1_1} + X_{1_2} + X_{1_3}\right) + \ldots + \left(X_{30_1} + X_{30_2} + X_{30_3}\right) \tag{18}$$

In (18), the i^th cluster with $i = 1,\ldots,c; c = 7$ shows the k number of common lecturers $k = 1,\ldots,n; n = 30$ and W_{ik}^{SC} constraints satisfied for X_{ik} common lecturer (k^th lecturer at i^th cluster). In (18), $Total_{const}^{SC}$ expresses all the constraints of common lecturers at each cluster per common lecturer. For example, $\left(X_{1_1} + X_{1_2} + X_{1_3}\right)$ means the feature of common lecturer 1 has been satisfied at cluster 1.

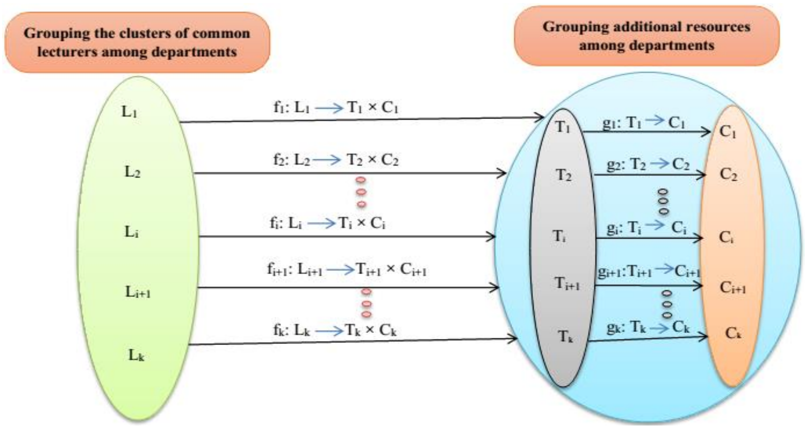


Fig. 6: Mapping the clusters of common lecturers in additional resources

Equation (19), represents the amount of competitiveness among clusters in terms of satisfaction percent of requirements, constraints and priorities of common lecturers among departments of each cluster, it means that we could find that at which i^th cluster which k^th common lecturer has more satisfied priorities and requirements over other common lecturers within each i^th cluster and other j cluster with $j = i+1, i+2, \ldots$. The numerator of this fraction must compute the satisfaction percent of each k^th common lecturer in terms of each i^th cluster and the obtain that percent over other j cluster and the denominator of this fraction must find the sum of whole satisfactions of each common lecturer at i^th cluster with whole dissatisfactions of each common lecturer at i^th cluster and then this iterates per j remained clusters so that the percent of real satisfactions of each cluster with their common lecturers would be obtained over whole satisfactions and dissatisfaction of per cluster and then the satisfaction percent of each cluster could be obtained over common lecturers and their allocation priority to the additional resources by dividing and averaging the obtained values of each cluster.

$$CTDS_2^{(i_j)} = \frac{\sum_{i=1}^{c} \left(W_i^{SC} \times i\right)}{\sum_{j=i+1}^{c} \left(Total_{Const_j}^{SC} \times j\right)} \times 100 = \frac{\sum_{Const=1}^{r} \sum_{k=1}^{n} \sum_{i=1}^{c} W_{ik}^{Const} \times i}{\sum_{Const=1}^{r} \sum_{j=i+1}^{c} Total_{Const_j}^{Const} \times j} \times 100 \tag{19}$$

In (19), $i = 1,\ldots,c$ is the number of clusters, W_i^{SC} is the satisfaction percent of common lecturers' constraints of i^th cluster and j also represents the number of other clusters in addition to i^th cluster where $j = i+1,\ldots,c$ and $c = 7$. In (19), the value of W_i^{SC} must be calculated in terms of the number of satisfied constraints for k^th common lecturer at i^th cluster over total number of i^th cluster's constraints for the common lecturers within this cluster. After obtaining a percent for each i^th cluster and common lecturers of those clusters, we could observe that which clusters have maximum satisfaction degree or minimum violation, so at first that cluster would have the priority of allocation and after reaching for instance to i^th cluster, now we must look for those common lecturer within i^th cluster whose satisfaction percent is the highest or they have minimum violation percent over his/her features and requirements and this is done upon (18).

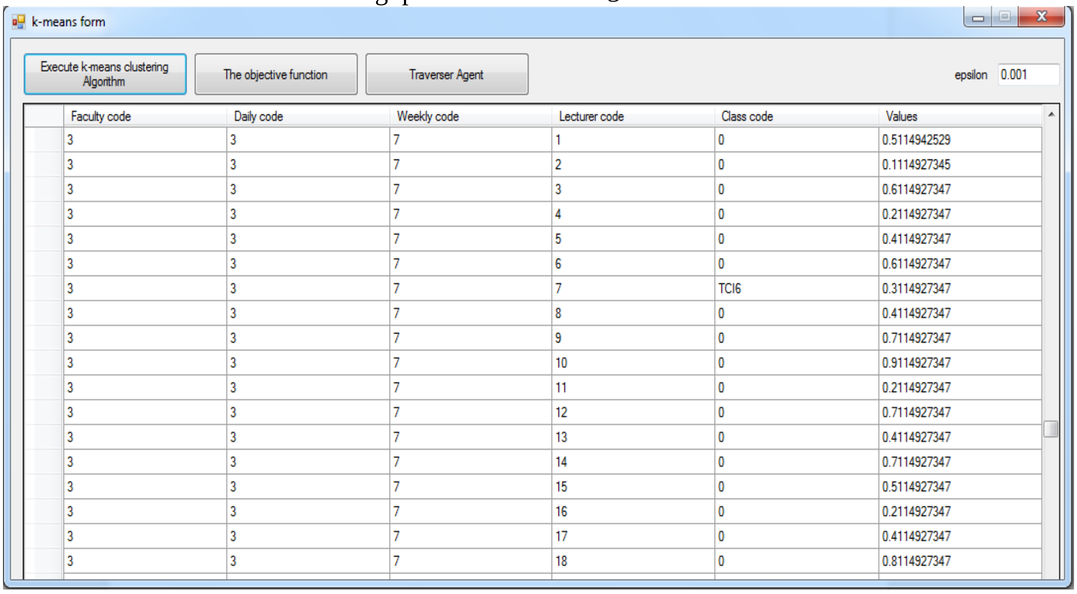
We could obtain the loss percent of additional resources among departments after clustering and mapping process per department based on (20).

$$ERW_A = \frac{a}{b} \times 100 \tag{20}$$

In (20), $ERW_A : (Extra\,Re\,sourcesWaste)_{After}$ means the loss of additional resources after clustering and mapping processes. Here, a represents the number of the remained additional resources of each department after allocation and b corresponds to the total number of existing resources at each department. To realize ERW_A equation, each department must apply its resources' allocation process to each common lecturer selectively (from the common lecturer himself/herself) and mandatory (from each department). The remained additional resources among departments equals to the subtraction of total number of departments' resources to the allocated resources by common lecturers and trainings of each department.

4.1 The performance of k-means clustering algorithm over dataset

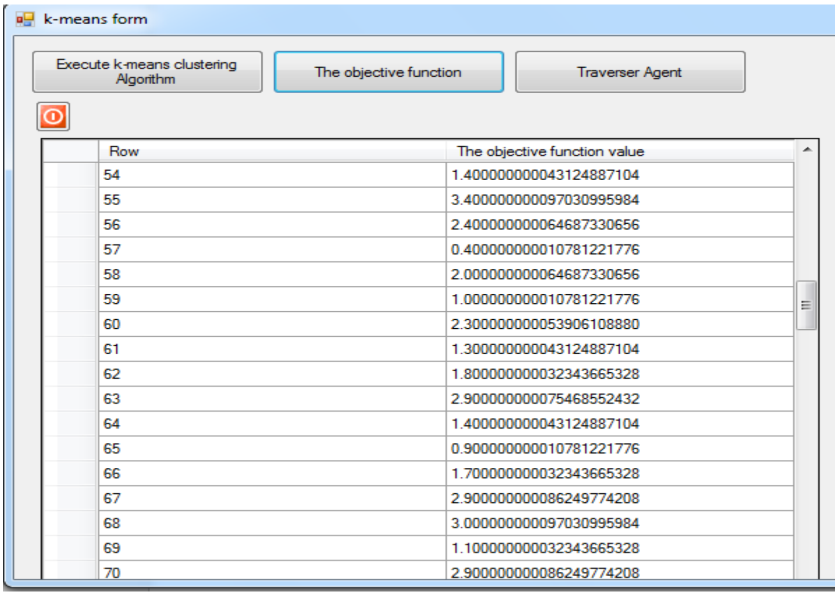
Based on the dataset presented in the first part of section 4, now we could test the k-means clustering algorithm on them. In Fig. 7, the k-means clustering algorithm based on the descriptions in sections 3.3.1 and following the sequence in applying the relations on k-means clustering algorithm compatible with common lecturers' time tabling problem have been shown. In Fig. 7, buttons Execute k-Means Algorithm, Traverser Agent and epsilon represent the performing of compatible k-means algorithm, the traversing agent and the value of parameter $\varepsilon = 0.001$, respectively. The six columns in Fig. 7, each one from left to right represent the faculty code, daily timeslot code, weekly timeslot code, common lecturer code, classroom code (theory-practical) and the computed values after pressing button Execute k-Means Algorithm with considering the value of epsilon=0.001. Button Traverser Agent in section 4.4 would present the way of traversing additional resources of faculties accompanied with mapping common lecturers to those additional resources. The column 6 which is the computed value is obtained after applying the rules of section 3.3.1 on the compatible k-means clustering algorithm.



Faculty code	Daily code	Weekly code	Lecturer code	Class code	Values
3	3	7	1	0	0.5114942529
3	3	7	2	0	0.1114927345
3	3	7	3	0	0.6114927347
3	3	7	4	0	0.2114927347
3	3	7	5	0	0.4114927347
3	3	7	6	0	0.6114927347
3	3	7	7	TCI6	0.3114927347
3	3	7	8	0	0.4114927347
3	3	7	9	0	0.7114927347
3	3	7	10	0	0.9114927347
3	3	7	11	0	0.2114927347
3	3	7	12	0	0.7114927347
3	3	7	13	0	0.4114927347
3	3	7	14	0	0.7114927347
3	3	7	15	0	0.5114927347
3	3	7	16	0	0.2114927347
3	3	7	17	0	0.4114927347
3	3	7	18	0	0.8114927347

Fig. 7: The result of applying k-means clustering algorithm

In Fig. 8, objective function in k-means clustering algorithm computed.



Row	The objective function value
54	1.40000000043124887104
55	3.40000000097030995984
56	2.40000000064687330656
57	0.40000000010781221776
58	2.00000000064687330656
59	1.00000000010781221776
60	2.30000000053906108880
61	1.30000000043124887104
62	1.80000000032343665328
63	2.90000000075468552432
64	1.40000000043124887104
65	0.90000000010781221776
66	1.70000000032343665328
67	2.90000000086249774208
68	3.00000000097030995984
69	1.10000000032343665328
70	2.90000000086249774208

Fig. 8: The result of objective function computed in k-means clustering algorithm

4.2 Comparison of adopted fuzzy c-means and proposed funnel-shape clustering algorithm with the k-means clustering algorithm adopted

In this section, we have shown the process of comparing k-means, fuzzy c-means and funnel-shape clustering algorithms in figures 9, 10 and 11, respectively and also provided a brief comparison as distinct for each faculty based on each 3 clustering algorithms in Fig. 12. In Fig. 12, we have shown a final pie chart in terms of common lecturers satisfaction percent based on each clustering algorithm.

In Fig. 9, the comparison result of k-means algorithm's satisfaction is shown for each 25 common lecturers among faculties as 3D (three dimensional). In this Fig., three length, width and height dimensions represent the common lecturer's code, the faculty code and the satisfaction percent of common lecturers, respectively.

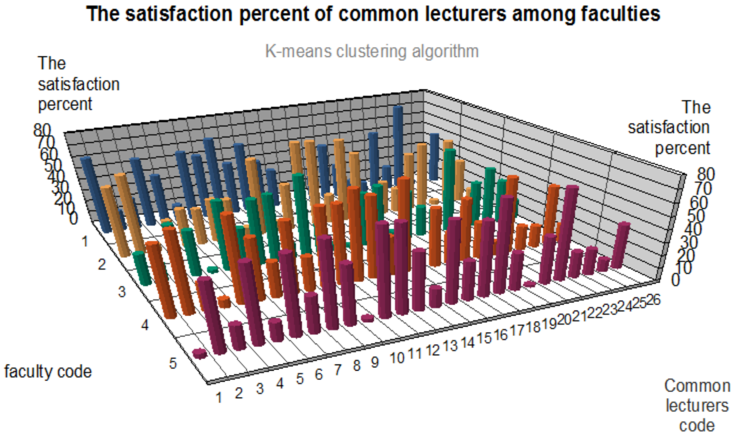


Fig. 9: The satisfaction percent of common lecturers based on k-means algorithm.

In Fig. 10, the comparison result of fuzzy c-means algorithm's satisfaction of each 25 common lecturers among faculties is shown as 3D (three dimensional). In this figure, three length, width and height dimensions represent the common lecturers' code, the faculty code and the satisfaction percent of common lecturers, respectively.

In Fig. 11, the comparison result of funnel-shape algorithm's satisfaction of each 25 common lecturers among faculties is shown as 3D (three dimensional). In this figure, three length, width and height dimensions represent the common lecturers' code, the faculty code and the satisfaction percent of common lecturers, respectively.

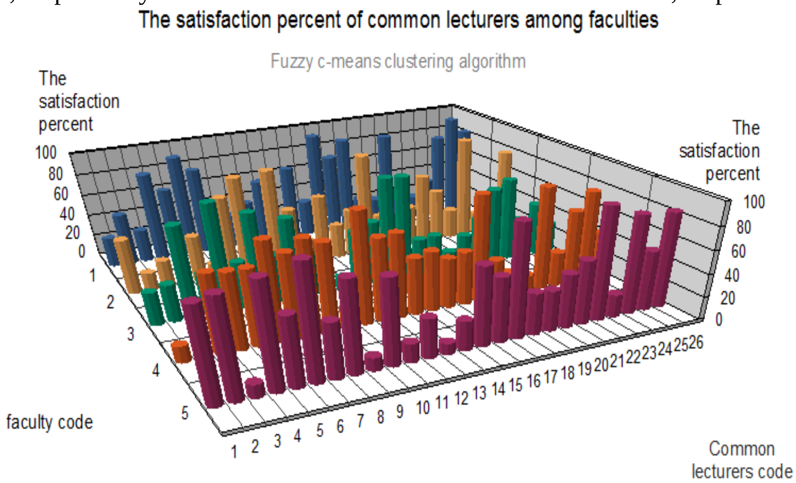


Fig. 10: The satisfaction percent of common lecturers based on fuzzy c-means algorithm

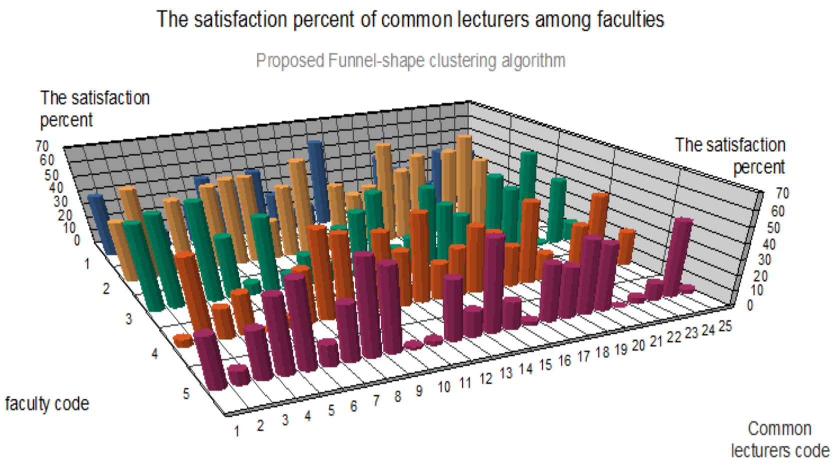


Fig. 11: The satisfaction percent of common lecturers based on funnel-shape algorithm

In Fig. 12, the minimum and maximum satisfaction percent of common lecturers among faculties have been shown for 5 faculties, 25 common lecturers corresponding to the dataset and 3 clustering algorithms. In the first five figures, the satisfaction percent of common lecturers is shown based on each clustering algorithm per faculty and finally the pie chart in the Fig. 12 shows the summary of satisfaction percent of common lecturers among faculties separately and in terms of clustering algorithms. The satisfaction percent of k-means, fuzzy c-means and funnel-shape clustering algorithms are as 28.19%, 38.6% and 33.2 %.

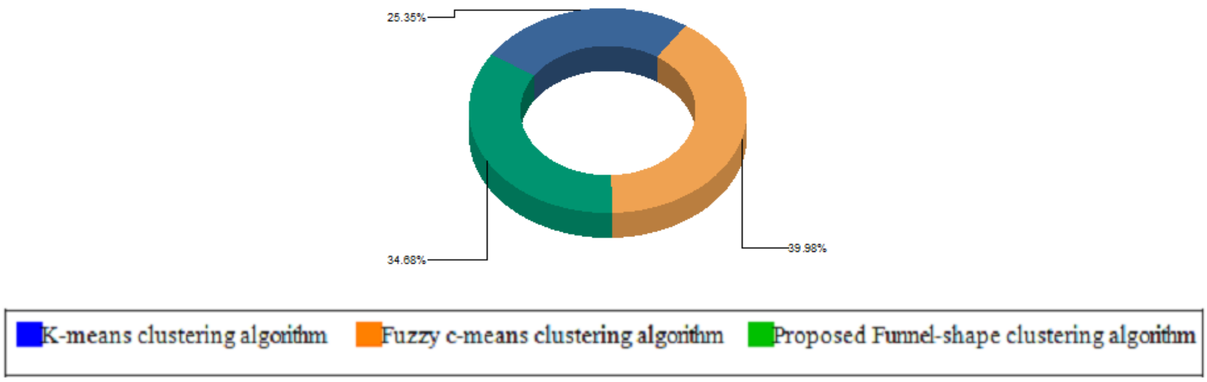


Fig. 12: The descending satisfaction percent of priorities of common lecturers among departments based on clustering algorithms

4.3 Traversing (additional) resources among departments and mapping the clusters of common lecturers by k-means clustering algorithm

Fig. 13 show the way of mapping the clusters of common lecturers to the additional resources of each 5 faculties by using k-means clustering algorithm.
In this shape, by clicking the button of deleting the allocated resources, all previously allocated resources per faculty are removed and by selecting the button of allocating the additional resources to the common lecturers, the act of emptying the stack of common lecturers' clusters list is done to map to the additional resources among faculties.
Since the assumptions related to the constraints and resources have been considered constant per faculty, so the allocation is done based on two selections where one is from the education (the related group) of each faculty and the other one is from the common lecturers among faculties.

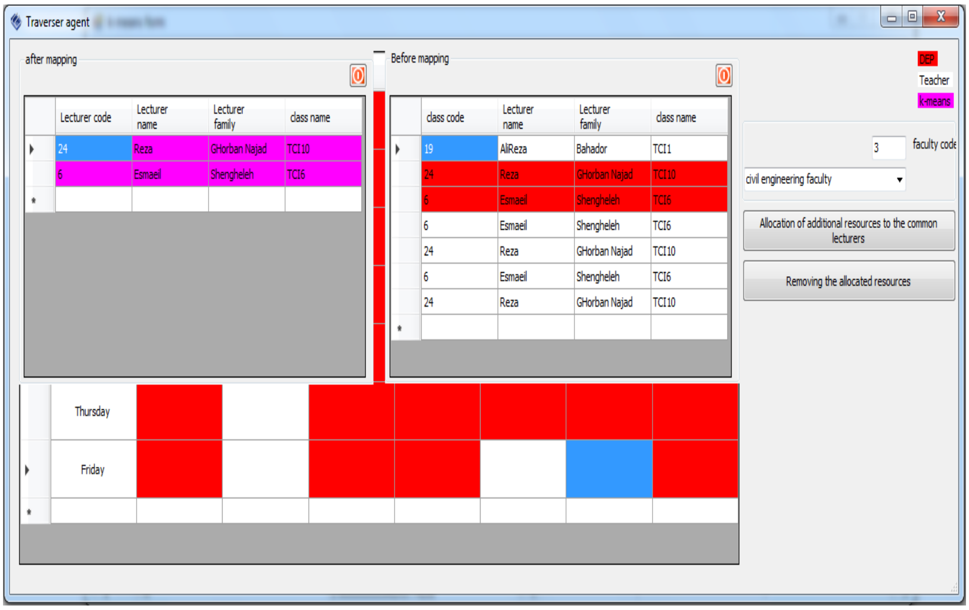


Fig. 13: Mapping the clusters of common lecturers in additional resources among departments with k-means algorithm

In Fig. 13, the red color shows the education (group) selections of each faculty, the white color represents the selections of each common lecturer, in Fig. 13 the purple color show the allocations of selections of each common lecturer to their constraints and priorities in k-means clustering algorithm after mapping process. Fig. 14 shows the additional resources loss percent per five faculties corresponding to each clustering algorithm. Table 1 shows the overall result of each three algorithms based on three clustering algorithms. However, here we could say that the first goal is to minimize the loss of additional resources of faculties for clustering algorithms from the maximum to minimum fuzzy c-means clustering (41.288%), funnel shape clustering (the proposed funnel) (32.55%) and k-means clustering (26.16%) and the second goal is to satisfy the priorities of common lecturers among faculties in a descending manner where for clustering algorithms from the maximum to minimum as fuzzy c-means clustering (38.6%), the proposed funnel clustering (33.2%) and k-means clustering (28.1%).

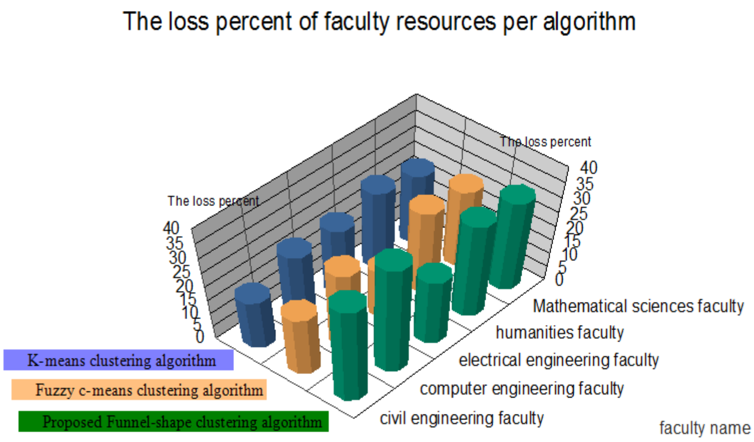


Fig. 14: Minimizing the loss of additional resources among departments through clustering algorithms

Table 1: Comparison of clustering algorithms based on research goals

Research goals	*Standard clustering*		*The proposed clustering*
	Fuzzy c- means clustering	*k-means clustering*	*The proposed funnel-shape clustering*
Loss minimization Faculties additional resources	41.288%	26.16%	32.55%
Descending satisfaction of common lecturers priorities	38.6%	28.1%	33.2%

5. Discussion

In this section, effects of the proposed method's advantages and disadvantages are discussed.

5.1 Disadvantages

1- Variability of lecturers' constraints and priorities in department where in the real context, it is not possible to satisfy all the requirements and priorities of involved events in a desirable extent and for this purpose a descending satisfaction is considered.
2- Limitation of appropriate and desirable resources in system to perform lecturers' timetabling process and traversing resources.
3- Not applying meta-heuristic and hybrid methods which leads to relative loss of efficiency of proposed algorithm in generating tables with primary timetabling ability within the existing agents in the system.

5.2 Advantages

1- Considering the priorities of lecturers specifically and their constraints in order to uniformly distribution over available resources.
2- In timetabling lecturers, most of their clear features are employed sufficiently.
3- Applying multi agent system based method to increase the autonomy of each department's timetabling where this autonomy prevents unplanned collisions and allocations among agents within distributed environment.

6. Conclusion

In this article, the obtained results from the CLTTP problem's purposes through the proposed approach include: 1- the proposed method results in a descending satisfaction from the priorities (soft constraints) of common lecturers among departments to allocate additional resources and 2- the loss of additional resources (unused) at each minimized department which represents the allocation of common lecturers to resources with an improving process. The future approach to solve UCTTP problem would be to work on multi agent based methods as a distributed architecture and apply modern syntactic and fuzzy meta-heuristic approaches where for example we can use meta-heuristic algorithms for two agents TA_i and MA in order to increase throughput in generating and improving time tables. In this problem we can use fuzzy c-means clustering algorithm by applying features weight learning (soft constraints of common lecturers) in generating more improved time tables based on common lecturers among departments where this algorithm could be executed after performing the process of mapping function f and transferring time tables to each agent (department). It must be noted that this method could be used to generate improved time tables in the first phase for each department locally. However, various types of events and resources' features within CLTTP problem could be considered in different kinds of clustering methods and various mapping methods could be used in such clustering approaches.

References

[1] Babaei, H., Karimpour, J., and Hadidi, A., "A survey of approaches for university course timetabling problem," Computers & Industrial Engineering 86, 2015, pp. 43–59.
[2] Feizi-Derakhshi, M. R., Babaei, H., and 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, 2012, pp. 307-321.
[3] 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.
[4] Gotlib, C. C., "The Construction of Class-Teacher TimeTables,", Proc IFIP Congress, Vol. 62, 1963, pp. 73-77.
[5] Asmuni, H., Fuzzy Methodologies for Automated University Timetabling Solution Construction and Evaluation, Ph.D. Thesis, School of Computer Science University of Nottingham, Nottingham, UK, 2008.
[6] Lewis, M. R., Metaheuristics for University Course Timetabling, Ph.D. Thesis, Napier University, Napier, UK, 2006.
[7] 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, USA, 2004.
[8] Srinivasan, S., Singh, J., and Kumar, V., "Multi-Agent based Decision Support System Using Data Mining and Case Based Reasoning," IJCSI International Journal of Computer Science Issues, 2011, Vol. 8, Issue 4, No 2.
[9] Obit, J. H., Landa-Silva, D., Ouelhadj, D., Khan Vun, T., and Alfred, R., "Designing a Multi-Agent Approach System for Distributed Course TimeTabling," 2011, IEEE.
[10] Wangmaeteekul, P., Using Distributed Agents to Create University Course TimeTables Addressing Essential Desirable Constraints and Fair Allocation of Resources, Ph.D. Thesis, School of Engineering & Computing Sciences Durham University, Durham, UK, 2011.
[11] Shatnawi, S., Al -Rababah, K., and Bani-Ismail, B., "Applying a Novel Clustering Technique Based on FP- Tree to University Timetabling Problem: A Case Study," 2010, IEEE.
[12] DeWerra, D., "An Introduction to TimeTabling," European Journal of Operational Research 19, 1985, pp. 151-162.
[13] Asham, G.M., Soliman, M.M. and Ramadan, A.R., "Trans Genetic Coloring Approach for Timetabling Problem," Artificial Intelligence Techniques Novel Approaches & Practical Applications, IJCA, 2011, pp. 17-25.
[14] Daskalaki, S., Birbas, T., and Housos, E., "An integer programming formulation for a case study in university timetabling," European Journal of Operational Research 153, 2004, pp. 117–135.
[15] Daskalaki, S., and Birbas, T., "Efficient solutions for a university timetabling problem through integer programming," European Journal of Operational Research 160, 2005, pp. 106–120.
[16] Khonggamnerd, P., and Innet, S., "On Improvement of Effectiveness in Automatic University Timetabling Arrangement with Applied Genetic Algorithm," 2009, IEEE.
[17] Alsmadi, O. MK., Abo-Hammour, Z. S., Abu-Al-Nadi, D. I., and Algsoon, A. , "A Novel Genetic Algorithm Technique for Solving University Course Timetabling Problems," 2011, IEEE.
[18] Mayer, A., Nothegger, C., Chwatal, A., and 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.
[19] Ayob, M., and Jaradat, G., "Hybrid Ant Colony Systems For Course Timetabling Problems," 2nd Conference on Data Mining and Optimization 27-28 October 2009, Selangor, Malaysia, IEEE, 2009, pp. 120-126.

[20] Jat N. S., and Shengxiang, Y., "A Memetic Algorithm for the University Course Timetabling Problem," 20th IEEE International Conference on Tools with Artificial Intelligence, IEEE, 2008, pp. 427-433.
[21] Alvarez, R., Crespo, E., and Tamarit, J. M., "Design and Implementation of a Course Scheduling System Using Tabu Search," European Journal of Operational Research 137, 2002, pp. 512-523.
[22] Aladag, C. H., Hocaoglu, G., and Basaran, A. M., "The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem," Expert Systems with Application 36, 2009, pp. 12349–12356.
[23] Tuga, M. Berretta, R. and Mendes, A., "A Hybrid Simulated Annealing with Kempe Chain Neighborhood for the university Timetabling Problem," 6th IEEE/ACIS International Conference on Computer and Information Science, 2007, ICIS.
[24] Shengxiang, Y., and Jat, N.S., "Genetic Algorithms with Guided and Local Search Strategies for University Course Timetabling," IEEE Transactions on Systems, MAN, and Cybernetics-PART C: Applications and Reviews, 2011, Vol. 41, No. 1.
[25] Abdullah, S., Burke, E.K., and McColloum, B., "An Investigation of Variable Neighborhood Search for University Course Timetabling," In The 2th Multidisciplinary Conference on Scheduling: Theory and Applications, NY, 2005, pp. 413-427.
[26] Kostuch, P., "The University Course Timetabling Problem with a Three-Phase Approach," In Lecture Notes in Computer science, pages 109-125, 2005, Springer-Berlin / Heidelberg.
[27] Shahvali Kohshori, M., and Saniee Abadeh, M., "Hybrid Genetic Algorithms for University Course Timetabling," IJCSI International Journal of Computer Science Issues, 2012, Vol. 9, Issue 2, No 2.
[28] Asmuni, H., Burke, E.K., and Garibaldi, J.M., "Fuzzy multiple heuristic ordering for course timetabling," The Proceedings of the 5th United Kingdom Workshop on Computational Intelligence (UKCI05), 2005b, pp 302-309.
[29] Golabpour, A., Mozdorani Shirazi, H., Farahi, A. kootiani, M., and H. beige, "A fuzzy solution based on Memetic algorithms for timetabling," International Conference on MultiMedia and Information Technology, IEEE, 2008, pp. 108-110.
[30] Chaudhuri, A., and Kajal, D., "Fuzzy Genetic Heuristic for University Course Timetable Problem," Int. J. Advance. Soft Comput. Appl., 2010, Vol. 2, No. 1, ISSN 2074-8523.

Hamed Babaei received the B.Sc. and M.Sc. degrees in computer engineering from Islamic Azad University (IAU), Iran, in 2007 and 2011, respectively. He is currently teaching in Department of Computer Engineering, Islamic Azad University of Ahar, Ahar, Iran. His current research interests include evolutionary algorithms, Meta-heuristic approaches, Distributed data base, Distributed systems, and their applications for timetabling problems.

Jaber Karimpour received the B.Sc, M.Sc. and Ph.D. degrees in computer systems from university of tabriz, Tabriz, Iran, in 1997, 2000 and 2009, respectively. He is currently teaching in Department of Computer Sciences, University of tabriz, Tabriz, Iran. His current research interests include theoretical Computer Science, Software Component Technology; Software and System specification, design and verification; Applied Mathematics in Software Engineering (Formal Methods); Temporal Logic; Modal Logic; Object Oriented Programming.

Sajjad Mavizi received the B.Sc. and M.Sc. degrees in computer engineering from Islamic Azad University (IAU), Iran, in 2008 and 2012, respectively. His current research interests include evolutionary algorithms, Meta-heuristic approaches, mobile ad hoc networks, vehicular ad hoc networks.