

An Automatic Course Scheduling Approach Using Instructors' Preferences

<http://dx.doi.org/10.3991/ijet.v7i1.1775>

Ibrahim Aljarah¹, Ayad Salhieh², Hossam Faris³

¹ North Dakota State University, Fargo, ND, USA

² Jordan University of Science & Technology, Irbid, Jordan

³University of Jordan, Amman, Jordan

Abstract—University Courses Timetabling problem has been extensively researched in the last decade. Therefore, numerous approaches were proposed to solve UCT problem. This paper proposes a new approach to process a sequence of meetings between instructors, rooms, and students in predefined periods of time with satisfying a set of constraints divided in variety of types. In addition, this paper proposes new representation for courses timetabling and conflict-free for each time slot by mining instructor preferences from previous schedules to avoid undesirable times for instructors. Experiments on different real data showed the approach achieved increased satisfaction degree for each instructor and gives feasible schedule with satisfying all hard constraints in construction operation. The generated schedules have high satisfaction degrees comparing with schedules created manually. The research conducts experiments on collected data gathered from the computer science department and other related departments in Jordan University of Science and Technology- Jordan.

Keywords—Data mining, Scheduling, Timetabling.

I. INTRODUCTION

Time synchronization has been extensively related to another term, which is the scheduling concept. The term scheduling has been intensively introduced; mainly in the operating system paradigm; nevertheless, scheduling concept has emerged in several other applications such as: client-server systems and aircraft systems. Variety of domains and areas require automatic timetable construction process such as education sector, courses timetabling and exam timetabling [3], sport activities (matches timetabling), hospitals applications (nurse rostering), and companies management (employees timetabling) [1]. Educational timetabling requires constructing automatic timetables because it contains many different events and resources, which require to be assigned to a specific time periods. Educational timetable constructions techniques are mainly divided into two main types that depend on the type of institution.

The courses timetabling which answer the question of how will the courses be distributed into some predefined time slots with regard to the existence of some constraints that manage the distribution process and other factors such as the students and instructors desires as well as the institution's legislation and ambitions might be taken into consideration during the distribution process [3].

On the other hand, examination timetabling class concerns with answering the question of how to distribute the university's exams into some predefined time periods and

fixed-size rooms, taking into consideration a set of constraints [3].

These constraints used in the timetabling construction process provide the frame of the established timetable. All types of constraints are classified into two main classes which are the hard constraints and soft constraints. Hard constraints affect the construction of the timetable and must satisfy the details of every factor in the timetable to get feasible timetable. Soft constraints are not directly employed during the timetable construction but it is desirable to apply these constraints to obtain a rigorous timetable [4]. The satisfaction degree of soft constraints poses a crucial problem for the development of the timetable.

The rest of the paper is organized as follows. Section 2 presents different timetabling problems and highlights the university courses timetabling problem. Section 3 presents the problem statement and discusses the problem solution. Section 4 shows some experimental testes as well as the performance analysis applied on the proposed system. Section 5 addresses the conclusion and final remarks in addition to the future work.

II. RELATED WORK

Timetabling and its related issues have been addressed intensively for the past few decades. In the year 1963, the first research has been revealed in automatic timetabling and initial ideas about how to schedule instructors with school classes [2]. Since that date, numerous publications have carried out and many models that addressed to solve automatic timetabling problems in many domains such as school timetabling, university timetabling, sport matches timetabling, nurse rostering, and employees timetabling [1].

In general, the approaches employed to solve timetabling problems are similar in their nature and are applicable and compatible. This section gives brief background for the techniques proposed and applied to solve the university courses timetabling (UCT) problem. The approaches proposed to solve the University Courses Timetabling (UCT) problem are categorized with regard to different criteria such as heuristics [4,9], mathematical and linear programming [21] techniques in many ways by the literature and divided in various types.

Classification for solving approaches classify into four types: heuristic-based approaches will be discussed along with some improvements such as meta-heuristic and hyper-heuristic approaches. Furthermore, reasoning approaches such as: case-based reasoning and constraint-based approaches are discussed. Mathematical and linear

programming approaches play core role in solving UCT problem [4].

A. Heuristic-based approaches

The Heuristic-based approaches use heuristic concepts and heuristic search to construct and define the solutions for many problems and give good results. Through the recent decades, there has been a heaviness of literature on heuristic approaches to solve timetabling problems and many researches discuss the heuristic topics in related field. Some heuristic approaches employ heuristic ordering where a heuristic is used to measure the difficulty of scheduling a particular course and solve conflicting between other courses [4,9]. These approaches order courses by using heuristics and then assign the courses sequentially into proper time slot; so that, courses in the period are free-conflict with each other [3,5,7]. Most approaches that used heuristic orderings are based on the graph coloring approaches [7] which convert problem to graphs, which contained vertices representing courses and edges represent conflicts between courses.

The free-conflicted courses (no edges between them) marked by the same color and different colors present different time slots [7,9]. In the literature many ordering heuristics used such as:

1. Largest Degree ordering heuristic (LD) which means the courses have more conflicts with other courses scheduled first because considered more difficult to schedule than other courses [7,9].
2. Largest Weighted Degree ordering heuristic (LWD) which presents updated version of the largest degree ordering heuristic. This ordering adds weights for each conflict using number of students in each conflict [3,7].
3. Saturation Degree ordering heuristic (SD) which ordering the courses achieved according the number of available feasible time slots for each course [7,9].
4. Largest Colored Degree ordering heuristic (LCD) based on LD, which mean the courses have more conflicts with other courses from the already assigned courses be scheduled first [3].

The main advantage for these orderings, it is easy to implement. After the courses ordered, variety of approaches can be used to choose the best time slot for each course.

Meta-heuristics approaches are considered improvements approaches on graph-colouring heuristics ordering, which begin with initial problem solution and employ some searching strategies to create more quality solutions. The quality of solutions depends on the type of the searching strategy and structure of the problem. The Meta-heuristics approaches are intensively used in the literature to solve the UCT problem by using some artificial intelligence searching algorithms such as: Simulated Annealing (SA) [15] Tabu Search (TS) [13], and Genetic Searching (GS) algorithms [17]. The main advantage for these searching algorithms is the efficiency through large problem spaces [4,10]. The SA is considered as the most meta-heuristic used in timetabling domain which forms iterative enhancement procedures that are designed to search for solutions that have more quality without being trapped at a local optimum [10]. Many algorithms used SA to improve initial solution such as the algorithm proposed by Elmohamed to solve UCT problem.

Simulated annealing has number of advantages such as easy to implement, applicable for most problems required optimization to get good solutions and easy to combine with many heuristics to improve results and have robust structure.

The disadvantages of SA method, it needs long time to get good solutions and must supply some parameters with awareness [16]. Another meta-heuristics used to solve timetabling problem is the Tabu Search (TS) method, which remembers the features of prior solutions to avoid visiting them again. This reduces the search space and gets results relatively quickly. The basic TS algorithm is started with some neighborhood timetables produced from current timetable, then selected neighborhood timetable which has minimum cost to become new timetable. In 1991, Hertz [12] proposed a number of techniques that incorporated the use of a Tabu Search in timetabling construction. Hertz [11] applied the method in more complex problems such as course scheduling with variable lecture duration doesn't know in advance [13,14].

B. Genetic approaches

The Genetic Searching (GS) algorithms are other meta-heuristics approaches, which employed to obtain high quality timetables. Many papers written in the literature employ and apply the genetic algorithms in their approaches to solve the timetabling problems such as [5].

In general, a genetic searching method starts by producing randomized timetables which present a parent population for the timetabling problem. After that, each generated timetable is converted to consistent timetable by eliminating courses that cause conflicting with other courses. Some initial timetables may be empty which no courses are scheduled. After that, selection criterion applied to choose timetables that used to get new parent population using genetic operators [18]. This operation repeated until the produced solution contains all scheduled courses and soft constraints satisfied with maximum satisfaction degree.

C. Reasoning approaches

Reasoning approaches are considered new methodologies in problem solving. Two types of reasoning approaches are applied on the UCT problem: Case-Based Reasoning (CBR) [5,20] approaches and Constraint-Based Reasoning approaches. Case-Based Reasoning (CBR) approaches are considered new methodologies in solving timetabling problems which use previous timetables and previous construction methodology in solving latest timetabling problems by using similarity measures. The big challenge for these approaches is a definition for similarity measures between timetables [5].

Constraint-based reasoning approaches treated UCT problem and modeling it as Constraint Satisfaction Problem (CSP) which is modeling as a set of variables that have finite domains controlled by a set of constraints. Each constraint is related to subset of variables and specifies the values for them. The consistent solution is free-conflict solution which is achieved if an assignment that doesn't violate any constraints related to the problem [20]. The satisfaction of all CSP constraints sometimes is impossible, because some precedence mechanism between constraints must employ to prefer constraint over another [19]. The main advantage for this method is very fast when it deals with small instances [19].

D. Mathematical Programming approaches

Mathematical Programming (MP) is considered elementary systems modeling which describe the relations between elements in the system by using mathematical way. Each MP method contains two elements variables which present the state of the problem and constraints which represent restrictions on the variables. The solution of the problem is assigned values for the variables and satisfying the constraints on these variables. Each generated solution has an objective function value which it is calculated for evaluating the solution. There are three types of the MP approaches are applied in UCT problem solving: Liner Programming (LP) approaches, Integer Programming (IP) approaches and Mixed Integer Linear Programming (MILP) approaches [21].

III. PROBLEM FORMULATION

Proposed approach will take specific formulation of university courses timetabling problem. The formulation for the problem reflects the structure of our approach.

A. Problem Components

The proposed approach contains four basic components (courses, periods, rooms and instructors) and each of them must define before starting in the model. Schedule have n courses C1, C2 ... Cn , and each course Ci is divided into k sections S1,S2,..., Sk , this depends on the number of students will enroll in the course, capacities of the rooms that reserved for each section and university administration decision.

University has j levels of courses L1, L2...Lj, which depend on the level year that have common students. The courses are categorized in two general categories depend on the type of course: compulsory courses and elective courses and each course section need number of lectures in the week; this number depends on the course class and duration for each lecture. The courses are categorized into three classes depended on the credit hour, class I needs 1 hour per week, class II needs 2 hours per week and class III needs 3 hours per week. University has r instructors I1, I2...Ir available to instruct the courses and each instructor knows the courses will be instructing in prior. University has d working days in the week D1, D2... Dd, distributed in multiple types depended on the cycle and courses repetition in the week. If that are 5 working days, the repetition of the lecture in the days takes these values: $\{(1,3,5),(2,4)\}$, $\{(1,3),(1,5),(3,5)\}$, $\{(1),(3),(5)\}$, this depends on the course class and duration for each lecture. University has t periods (time slots) P1, P2 ... Pt. each period is specified by start time and end time in the day, and each type of days (cycle days) has different time slots compared to other types.

The day repetition type that have (1, 3, 5) repetitions contains time slots T1, where each period gives 1 hour duration. The day repetition type that has (2, 4) repetitions contains time slots T2, where each period takes 1.5 hours duration. The day repetition type that has (1, 3), (1, 5), (3, 5) repetitions contains time slots type T1, where each period takes 1 hour duration. The day repetition type that has (1), (3), (5) repetitions contains time slots type T1, where each period takes 1 hour duration.

Each course section needs classroom from predefined reserved classrooms that are satisfied type of the course. University has m classrooms R1, R2... Rm classified in x

TABLE I.
UNITS FOR MAGNETIC PROPERTIES

Symbol	Conversion from Gaussian and CGS EMU to SI ^a
HC1	for all Ci, $F(i,s,j)=1$, $i \in \{C1,C2,..Cn\}$, $s \in \{S1,S2,..Sk\}$, $j \in \{D1,D2,..Ds\}$.
HC2	for all Ci, $L(i,s,w)=1$, $i \in \{C1,C2,..Cn\}$, $s \in \{S1,S2,..Sk\}$, $j \in \{p1,p2,..pt\}$.
HC3	for all Pi, total of lectures $\leq m$, m: number of classrooms in related time slot.
HC4	for all Ii, total of lectures in each p =1 {one lecture at the same time.}.
HC5	for all types of rooms $V(TRi)=Ei$, Ei fixed number present number of classrooms have TRi classroom type, i: classroom type.
HC6	for all Ci, $U(i,s,j)=1$, $i \in \{C1,C2,..Cn\}$, $s \in \{S1,S2,..Sk\}$, $j \in \{R1,R2,..Rm\}$.
HC7	for all Ri, total of lectures in each Ri =1 {one lecture at the same time.}.
HC8	for all Ci, Ci room type=TRj, i $\{C1,C2,..Cn\}$, $j \in \{R1,TR2,..TRx\}$.
HC9	for all Ri, $Ns \geq Capacity(Ri)$, i $\{R1,R2,..Rm\}$, Ns : Number of sections

multiple types TR1, TR2... TRX. This is depending on the room equipments such as classroom, lab, hall, etc. Classrooms are distributed on multiple buildings Bul1, Bul2, etc.

B. Constraints Formulation

The constraints used in our solution are classified into two classes:

1. Hard Constraints: these constraints must be satisfied in the schedule in order to consider it feasible. Also, the hard constraints are applied in the whole timetable. Table I shows nine hard constraints that used in our schedule construction.
2. Soft Constraints: these constraints are desirable to be satisfied by courses in the schedule in order to consider it more suitable. The satisfaction of such constraints is not obligated to say that the schedule is feasible since most of them are preferences for the problem components. There are three types of soft constraints, Time-based Soft Constraints (TSC), Day-based Soft Constraints (DSC) and Room-based Soft constraints (RSC).

IV. PROPOSED SOLUTION

The materialization of any soft constraint adds more feasibility to courses schedule and get more desirable schedule. The satisfaction of the soft constraint is measured by the degree of satisfaction which means the satisfaction of the soft constraint it is not 0 or 1, may be satisfied 50% or 60% in courses schedule. Preferences constraints are considered type of soft constrains that expressed the people desires concerned with schedule construction such as instructors, students and university administration. In the most universities the instructor preferences are considers the first factor used to measure

the quality of the courses schedule. In this research, we will propose new technique to collect the instructor preferences by using the previous courses schedules. After that, we will employ this technique to construct courses schedule by developing a day courses schedule firstly, then we will distribute the courses in multiple time terms in each day type.

As mentioned in the previous sections, the constraints are defined in timetable that have different satisfaction degree, hard constraint must be satisfied to get feasible courses timetable, any violation to these defined constraints in the final solution lead to unfeasible courses schedule, the satisfaction of the hard constraint is binary zero or one, zero if the constraint not satisfy and one otherwise. Another type of the constraints is the soft or desirable constraints, some of the desirable constraints may be satisfied in the courses schedule and other constraints may be not.

The automation of schedule construction can take some information from the previous schedules by mining instructor preferences that present new soft constraints for the new constructed schedule. The proposed approach is divided into multiple phases, mining phase and schedule construction phase.

A. Mining Instructors' Preferences

Most universities are considered the instructors' preferences satisfaction in the courses schedules construction is the main challenge and big effector to compare the schedules quality. The schedules that have high instructor satisfaction degree better than the schedules have low satisfaction degree. The Instructor Satisfaction Degree (ISD) means the average number of the soft constraints (preferences) satisfied in the schedule construction over all the soft constraints exist in the problem. ISD is considered the main measure to specify the schedule quality in this research.

Data mining [6] is a good tool to discover and extract information from a huge amount of data. Every semester, the university administration establishes schedules manually by using the available information from the instructors and the final constructed schedule reflects some instructors' preferences collected by using questioners that give to instructors every semester. In this paper we will use new technique to collect the instructor's preferences without referring to instructor every semester by using mining algorithm to extract these preferences from historical schedules from prior years. The instructor's preferences are divided into two categories: predefined preferences category and mined preferences category. Predefined preferences are obtained by asking new instructors come to the university in order to specify their preferences. Mining preferences are obtained by association rule mining algorithm [6] that specifies the frequent instructor preferences in the previous schedules. Predefined and mined preferences are classified into three types: day preferences, time preferences and building/classroom preferences. Time preference has three types: Morning (M) time term, Afternoon (A) time term and Evening (E) time term.

The resulting preferences constitute as rules that are used in schedule construction two types of rules:

(One to One) rules associate the instructor name with preferred day, time term and building.

Rule 1: Instructor name \Rightarrow Repetition day

Rule 2: Instructor name \Rightarrow Time term

Rule 3: Instructor name \Rightarrow Building

(Two to One) rules associate the instructor name and specific course with prefer day and time term.

Rule 4: Instructor name \Rightarrow specific course \Rightarrow repetition day

Rule 5: Instructor name \cap specific course \Rightarrow time term

For example:

Instructor (A) \Rightarrow (1, 2, 3) repetition day (for all courses)

Instructor (A) \cap Course (C1) \Rightarrow (1,2,3) repetition day (for C1 only).

In addition, the benefit behind mining preferences is to get information about the laws of the specific institute (university) that discriminate it over other institutes, such as working days, time slots used and types of courses, some constraints over classrooms and courses sections, number of sections for any course are used in previous schedules.

We use standard Apriori [6] association rule algorithm to extract preferences from previous schedules. Figure 1 represents instructor's preferences extraction algorithm.

Table 2, Table 3, Table 4, Table 5 and Table 6 represent some examples for Rule 1, Rule 2, Rule 3, Rule 4 and Rule 5 respectively, after applying extract instructors' preferences algorithm from some previous schedules.

(1)	Instructor_Preferences_Extraction_Algorithm (schedule, instructors, courses)
(2)	For each schedule do
(3)	Standard_Apriori_Association_Rules (schedule)
(4)	For each instructor do
(5)	Find frequent preferences (instructor, day)
(6)	Find frequent preferences (instructor, time term)
(7)	Find frequent preferences (instructor, building)
(8)	For each course do
(9)	Find frequent preferences (instructor, day)
(10)	Find frequent preferences (instructor, time term)
(11)	Loop
(12)	Loop
(13)	Loop

Figure 1. Instructors' Preferences Extraction Algorithm.

TABLE II.
REPETITION DAY – MINING PREFERENCES (RULE 1)

Instructor	Repetition Day	Frequency
1	(1,3,5)	5
2	(2,4)	12
3	(1,3,5)	14
4	(2,4)	11
5	(2,4)	9
6	(2,4)	16
7	(1,3,5)	13

REGULAR PAPER
AN AUTOMATIC COURSE SCHEDULING APPROACH USING INSTRUCTORS' PREFERENCES

TABLE III.
TIME TERM – MINING PREFERENCES (RULE 2)

Instructor	Repetition Day	Frequency
1	M	10
2	E	8
3	A	6
4	A	14
5	E	5
6	M	10
7	A	20

TABLE IV.
BUILDING – MINING PREFERENCES (RULE 1)

Instructor	Repetition Day	Frequency
1	Building1	11
2	Building4	6
3	Building5	7
4	Building2	5
5	Building1	6
6	Building3	7
7	Building2	6

TABLE V.
COURSE AND REPETITION DAY - MINING PREFERENCES (RULE 4)

Instructor	Course	Repetition Day	Frequency
1	C1	(1,3,5)	4
2	C2	(1,3,5)	3
3	C3	(1,3,5)	2
4	C4	(2,4)	3
5	C5	(2,4)	6
6	C6	(1,3,5)	4
7	C7	(2,4)	7

TABLE VI.
COURSE AND TIME TERM - MINING PREFERENCES (RULE 5)

Instructor	Course	Repetition Day	Frequency
1	C6	M	2
2	C2	M	3
3	C3	A	4
4	C1	M	6
5	C1	E	7
6	C2	A	4
7	C3	E	5

B. Courses Schedule Construction Phase

Courses schedule construction phase forms the main challenge in UCT problem solving. The instructor preferences are used from previous phase to form the structure of the schedule construction algorithm. The resulting schedule complies with instructors by increasing ISD. Figure 2 shows the schedule construction stages.

C. Determination the Number of Sections Stage

Number of sections and capacities for each course are obtained from the predefined number of students expected to enroll for each course and from the available information in the previous schedules. Some predefined constraints may be restricted number of the sections in some courses. Initially, the number of sections for any course depends on the average of the classrooms capacities and the number of students enrolled for the each course.

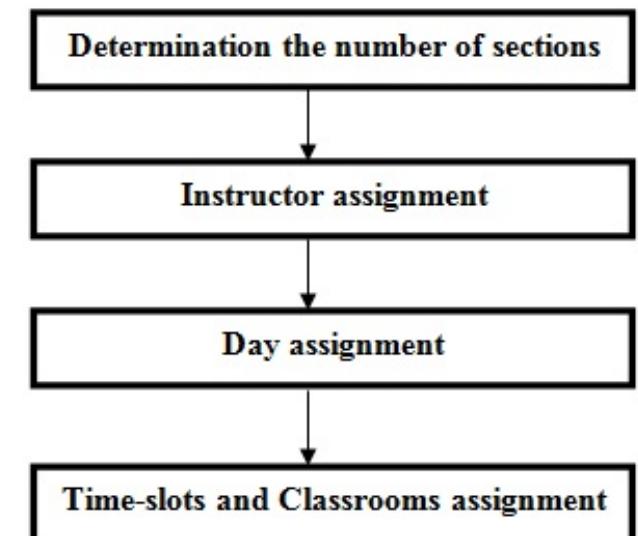


Figure 2. Courses schedule construction stages.

Through the remaining courses schedule construction stages, the number of sections for each course will be changed, this depend on the classroom chooses and its capacity.

D. Instructor Assignment Stage

Assigning the instructors to courses that guaranteed maximizing satisfaction of preferences for each instructor. The instructors' assignments are done by using priority measure for each instructor. Each course may be instructed by multiple instructors. The instructor selection is accomplished by using available information from the departments with some predefined information.

E. Day Assignment Stage

The day assignment stage is a new stage in the courses timetabling, our approach separates the day assignment and time slot assignment. The aim of this operation to reduce number of the courses will schedule in each time slot, this leads to reduce number of the conflicts between problem components.

Before day assignment stage started, the courses must be ordered accordingly which course is more important than others, the general importance mechanism is adopted in the proposed method to order the courses forms as follows:

- item Compulsory courses ordered first, then elective courses.
- item Courses for the four year level ordered first then third year level and soon.
- item Class III credit hour courses ordered first, then Class II, then Class I.

After this stage the TH1 Constraint is satisfied for any course section and all the day preferences constraints are satisfied.

F. Time slots/Classrooms assignment stage

The time slots and classroom assignment stage is considered core step in the schedule construction because the satisfaction of the time and room constraints will be checked and conflicted resolving between courses sections is done in this step. Figure 3 shows illustrating the basic steps in the time slots and classroom assignment stage.

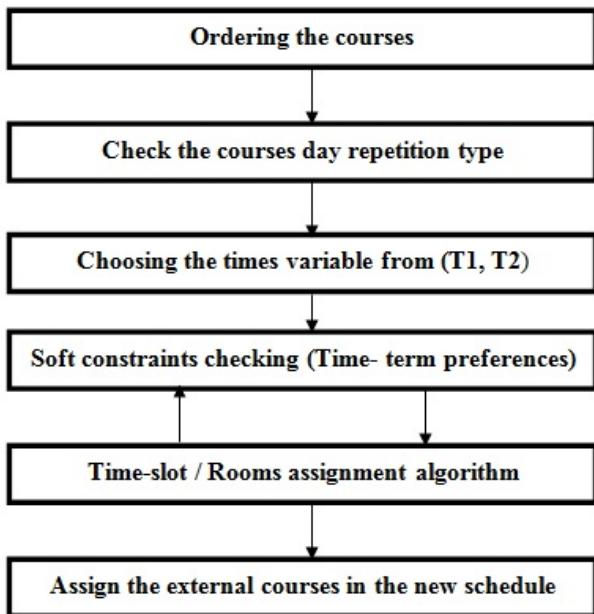


Figure 3. Time- slots and classroom assignment stage.

In Ordering courses stage, courses are scheduled in accordance with the difference of importance in timetable, because this, we will use the mechanism stated in the day assignment stage and use their measures to order the importance of the courses. Compulsory courses sections ordered first, then elective courses. Courses for the four year level ordered first then third year level. Also, Class III courses ordered first, then Class II, then Class I.

Checking the courses day repetition type stage is tested by scanning the output for the previous stage (Days assignment) to get the days repetition types. Choosing the times variable from (T1, T2) stage includes selecting time variable depends on the previous sub-stage and its constraints will be done. The checking soft constraints is also having some ordering depending on the problem component that has constraint; the ordering of checking soft constraints is done by checking soft constraints associated with the courses conflicts then checking instructor soft constraints (instructor preferences) then checking general soft constraints.

After this stage, every course has time term depended on the instructor preferences. Each course should have priority value which calculated using new function called Priority Function (PF). Courses will be ordered in each time term depend on apriority values. The priority function employed in the approach takes three measures to calculate: Course type, course level and course credit hour class. Every measure takes ratio from a whole priority function value. The following example explains how to calculate the priority function: Assume we have two courses: Course A, Course B. Course A and Course B are compulsory courses. Course A located in Level 3 and Course B located in Level 1. Course A classified as Class I course and Course B classified as Class III. PF (Course A) = $50\% + 12.5\% + 6.26\% = 68.76\%$ PF (Course B) = $50\% + 3.13\% + 25\% = 75.13\%$ Course B has more precedence than Course A and ordered before it.

Timeslots/Rooms assignment algorithm works after all courses sections are distributed on the day repetition types and time terms, every course should know which specific

time slot will be taken. Timeslots/rooms assignment algorithm consists in general into two stages:

- **Stage I:** generation free courses schedule such as, each line of schedule contains seven fields; Day, time slot, room, course_id, inst_id, reserved and conflict_timeslot. Course_id, inst_id, reserved and conflict_timeslot assigned zeros initially. For example, suppose we have two types of repetition days TD1 and TD2, TD1 contains 4 time slots and TD2 have 3 timeslots, number of available rooms in all buildings 4 rooms. The result free schedule for this example contains $4*4=16$ rows for TD1 and $3*4=12$ rows for TD2. The number of all rows in the free schedule is equal 28 lines.
- **Stage II:** Each course will be given time slot using the following ordered scenario:
 - Step 1: instructors seek for preferred day, preferred time term and preferred building at the same time for their courses in the free schedule. The time slot available for the course section selected in the free schedule must be checked to avoid conflicted with other timeslots have the same instructor. Figure 4 presents solving conflicting between instructors' courses algorithm.
 - Step 2: instructors seek for preferred day and preferred time term at the same time for their courses in the free schedule (without concern with preferred building).
 - Step 3: instructors seek for preferred day and preferred building at the same time for their courses in the free schedule. (Without concern with preferred time term). This step consists work with the following rules:
 - * IF the preferred time term is morning, instructor seeks for afternoon time slot.
 - * IF the preferred time term is afternoon, instructor seeks for evening time slot.
 - * IF the preferred time term is evening, instructor seeks for afternoon time slot.
 - * IF time slot unavailable until now:
 - * IF the preferred time term is morning, instructor seeks for evening time slot.
 - * IF the preferred time term is afternoon, instructor seek for morning time slot.
 - * IF the preferred time term is evening, instructor seek for morning time slot.
 - Step 4: instructors seek for preferred time term and preferred building at the same time for their courses in the free schedule. (Without concern with prefer day).
 - Step 5: instructors seek for preferred day for their courses in the free schedule. (Without concern with preferred time term and preferred building).
 - Step 6: instructors seek for preferred time term

for their courses in the free schedule. (Without concern with preferred day and preferred building).

- Step 7: instructors seek for preferred building for their courses in the free schedule. (Without concern with preferred day and preferred time term).
- Step 8: instructors seek for any available time slot / room in the free schedule.

The steps 5, 6 and 7 for timeslots/rooms assignment algorithm the same as other steps with applied on individual preference (Day, Time term and Building respectively). The last step in the algorithm assigns remaining unscheduled to any available time slot / room in the free schedule. The output schedule satisfies all hard constraints mentioned previous and satisfy maximum number of soft constraints.

	Solve_conflict_between_instructors_courses (found,free_schedule,count_row,inst(si))
(1)	For count_row2=1 to no_of_rows do
(2)	if free_schedule(count_row).t_slot = free_schedule(count_row2).t_slot and free_schedule(count_row).day = free_schedule(count_row2) and free_schedule(count_row2).inst = inst(si) then
(3)	found=true
(4)	loop

Figure 4. Conflicts solving between courses of instructors algorithm.

V. EXPERIMENTAL RESULTS

This section introduces some experiments applied on the developed system using real datasets obtained from the Computer Science Department and Computer Information Systems Department at the Jordan University of Science and Technology. The previous history for the CS and CIS Departments' schedules used to get the instructors' preferences using the data mining algorithm to generate the frequent preferences for each instructor in previous years. Our system takes all available previous schedules used in 2003, 2004, 2005, 2006, 2007 years as a previous history.

The developed system has been implemented using the visual C++ 6.0 programming language. The experiments are run on Pentium M machine with a clock rate of 1600 MHz and 512 Mbytes of main memory. The main concern in this research is to maximize instructor satisfaction for the courses schedule. The evaluation of the system must be assured if each instructor gets his preferences in high degree. The own major used to evaluate constructed schedule called Instructor Satisfaction Degree (ISD) used which means how many preferences constraints satisfied in the final schedule from the whole constraints expect each instructor will be satisfied. Equation 4.1 specifies the ISD for the instructor and Equation 4.2 specifies the ISD for the whole schedule.

Multipliers can be especially confusing. Write "Magnetization (kA/m)" or "Magnetization (10^5 A/m)." Do not

write "Magnetization (A/m) $\times 1000$ " because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

$$ISD(inst) = \frac{CSI}{K} * 100\% \quad (1)$$

Where, CSI: number of satisfied constraints for the instructor.

K: number of all constraints for instructor.

$$ISD(inst) = \frac{\sum_{i=1}^n CSI_i}{\sum_{i=1}^n Ki} * 100\% \quad (2)$$

Where, CSI_i : number of satisfied constraints for the instructor i.

n: number of instructors.

K_i: number of all constraints for instructor i.

We performed several experiments using courses that were taken by Computer Science (CS) and Computer Information Systems (CIS) Departments at Jordan University of Science and Technology. Experiment One divided into two parts, in the first part will be generate new schedule for Winter Semester of year 2005/2006 by using mining preferences from previous schedules only and the second part by using predefined preferences collected from instructors.

Experiment Two consists of two parts, in the first part will be generate new schedule for Fall Semester of year 2006/2007 by using mining preferences from previous schedules In the second part of Experiment two, the same schedule will be generated using the previous schedules used in the first part with adding the output schedule for 2005/2006 - Winter Semester from first part of Experiment One.

We have chosen 34 instructors in experiment one, the total number of courses was 38, out of which 29 courses are compulsory courses while the remaining 9 courses are elective courses distributed initially in 110 sections. The number of rooms available for the departments equal 15 rooms in variety types distributed into 4 buildings. The instructors will be expressed by using identifier numbers from 1 to 34. The courses will be used with the expected number of students will be registered in each course and initial number of sections for each course when the average capacity of the all rooms equal 40 students are given.

The approach extracts the instructor preferences by calculating the frequent preferences for each instructor using the Apriori algorithm that is used to mine these preferences from the previous schedules applied in last years. The mining operation discover three types of preferences: preferred day, preferred time and preferred building. The approach extracts these preferences for each instructor and applied in schedule construction. All instructor preferences used in this experiment collected from mining without any predefined preferences.

Schedule generated using our system have high Instructor Satisfaction Degree (ISD) for the instructors by comparing it with the existed schedules created manually that have the same year and the same semester. The differences between manual schedule and our generated schedule showed in the Table 7. We note in the Table 7 number of initial sections reduced by 13 sections; this lead to decrease number of instructors and utilize classrooms

using. The instructors' satisfaction degree when satisfying the day, time and building preferences at the same time better than schedule created manually. ISD increased when take in consideration day/time preferences and day/building preferences.

In the second part of the experiment, system will use predefined preferences collected from the instructors. Our system add more flexibility to deal with instructors have new preferences and deal with new instructors in the department. We will use the same university courses and rooms codes to accomplish this work.

The Instructor Satisfaction Degree (ISD) for the instructors for the generated schedule is larger than satisfaction degree for schedules created manually. The predefined preferences satisfied in the generated schedule in high ratio. The differences between manual schedule and our generated schedule showed in the Table 8. We note in the table 8 the number of initial sections reduced by 14 sections. The instructors' satisfaction degree when satisfying the day, time and building preferences at the same time better than schedule created manually. ISD increased when take in consideration day/time preferences and day/building preferences.

In the first part of the Experiment Two will be generate new schedule for Fall Semester of year 2006/2007 by using mining preferences from previous schedules then we will calculate the satisfaction degrees and will compared with the manual schedule obtained from the departments. The schedule generated in the using our system has high Instructor Satisfaction Degree (ISD) for the instructors by comparing it with the schedules created manually. The differences between manual schedule and our generated schedule showed in the Table 9.

In the second part of Experiment two, the same schedule will be generated using the previous schedules used in the first part with adding the new schedule 2005/2006 - Winter Semester to previous schedules from first part of Experiment One. Because the generated 2005/2005 - Winter Semester schedule has high satisfaction degree, this lead to generate 2006/2007 Fall semester more applicable and will increase satisfaction degree for it.

In this part of experiment the addition of new automatic schedule to previous schedules increased and satisfaction degrees better than schedule created manually. Table 10 shows the updated satisfaction degree for schedule 2006/2007-Fall Semester.

The main difference between two schedules is the number of sections for each course. The number of sections in the output schedules from our system less than number of sections in the manual schedule. The other difference, the manual schedule doesn't take instructor preferences by the way allow each instructor get the preferred times, preferred days and preferred building. But the generated schedules give these results in high degrees.

VI. CONCLUSIONS AND FUTURE WORK

Automatic Timetable construction is considered as one of the scheduling issues. This research contributes a new automatic courses scheduling and timetabling system using the previous history and previous schedules to extract preferences for each instructor; this system uses a hyper approach using Aprori mining algorithm \cite{key-6} along with available information to extract hard and soft constraints. Such constraints are used to confine the

scheduling process and to identify the tasks' priorities. The research aimed at solving the problems encountered in every semester by finding an automatic system for courses schedules and get high satisfaction degrees for each instructor; it is expected that the developed system will reduce effort and time for the department's workers who are involved in making these schedules.

We have only tested our system on Computer Science department courses; it will be interesting to see the performance of our system on some other departments courses. In our system we used the preferences for instructors. We may achieve better performance by using students preferences with instructor preferences; this is left as future work.

TABLE VII.
SATISFACTION DEGREES FOR 2005/2006 -WINTER SEMESTER
(MINING PREFERENCES)

Comparing Factor	2005/2006 Winter Semester (Manual)	2005/2006 Winter Semester (Automatic)
No of the initial sections	110	110
No of sections after schedule generated	110	97
ISD for Day, Time and Building preferences (ISDDTB)	0.16	0.82
ISD for Day and Time preferences (ISDDT)	0.32	0.94
ISD for Day and Building preferences (ISDBB)	0.37	0.88
ISD for Time and Building preferences (ISDTB)	0.22	0.83

TABLE VIII.
SATISFACTION DEGREES FOR 2005/2006 -WINTER SEMESTER
(PREDEFINED PREFERENCES)

Comparing Factor	2005/2006 Winter Semester (Manual)	2005/2006 Winter Semester (Automatic)
No of the initial sections	110	110
No of sections after schedule generated	110	96
ISD for Day, Time and Building preferences (ISDDTB)	0.16	0.67
ISD for Day and Time preferences (ISDDT)	0.32	0.76
ISD for Day and Building preferences (ISDBB)	0.37	0.88
ISD for Time and Building preferences (ISDTB)	0.22	0.66

TABLE IX.
SATISFACTION DEGREES FOR SCHEDULE 2005/2006 -FALL
SEMESTER

Comparing Factor	2005/2006 Winter Semester (Manual)	2005/2006 Winter Semester (Automatic)
No of the initial sections	120	110
No of sections after schedule generated	120	106
ISD for Day, Time and Building preferences (ISDDTB)	0.17	0.80
ISD for Day and Time preferences (ISDDT)	0.33	0.88
ISD for Day and Building preferences (ISDBB)	0.37	0.91
ISD for Time and Building preferences (ISDTB)	0.22	0.81

REGULAR PAPER
AN AUTOMATIC COURSE SCHEDULING APPROACH USING INSTRUCTORS' PREFERENCES

REFERENCES

- [1] Burke E, Jackson K, Kingston J, and Weare R. Automated University timetabling: the state of the art. *The Computer Journal*. 1997;40: 565-571. <http://dx.doi.org/10.1093/comjnl/40.9.565>
- [2] Gotlieb C. The construction of class-teacher timetables, in the proceedings of Popplewell Conference, IFIP congress, North-Holland. 1963: 73-77.
- [3] Carter M and Laporte G. Recent Developments in Practical Exam Timetabling. In the proceedings of 1st International Conference on the Practice and Theory of Automated Timetabling ,UK.1996: 1153; 3-21.
- [4] Carter M and Laporte G. Recent Developments in Practical Course Timetabling. In The proceed of the Practice \& Theory of Automated Timetabling conference, UK. 1998; 3-19.
- [5] Burke E and Petrovic S. Recent Research Directions in Automated Timetabling. *European Journal of Operational Research*, UK. 2002: 140 (2); 266-280. [http://dx.doi.org/10.1016/S0377-2217\(02\)00069-3](http://dx.doi.org/10.1016/S0377-2217(02)00069-3)
- [6] Agrawal R, Imielinski T and Swaim A. Mining Association Rules between Sets of Items in Large Databases. In proceedings of SIGMOD conference. 1993: 207-216.
- [7] Burke E, Meisels A, Petrovic S, and Qu R. A graph-based hyper heuristic for timetabling problems. *The European Journal of Operational Research*. 2006: 176 (2); 177-192.
- [8] Carter M. A comprehensive course timetabling and student scheduling system at the University of Waterloo. In the Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling ,UK. 2000: 48-59.
- [9] Burke E, Elliman D and Weare R. A University Timetabling System Based on Graph Colouring and Constraint Manipulation. *The Journal of Research on Computing in Education*. 1993: 27 (1); 1-18.
- [10] Rossi D, Sampels O, Chiarandini M, Knowles M, Manfrin J, Mastrolilli M, Paquete M and Paechter L. A comparison of the performance of different metaheuristics on the timetabling problem. In the Proceedings of International Conference on the Practice and Theory of Automated Timetabling, UK. 2002: 329-354.
- [11] Hertz A. Finding a feasible course schedule using tabu search. *Discrete Applied Mathematics journal*.1992: 35 (3); 255-270.
- [12] Hertz A. Tabu search for large scale timetabling problems. *European Journal of Operations Research*. 1991: 54; 39-47. [http://dx.doi.org/10.1016/0377-2217\(91\)90321-L](http://dx.doi.org/10.1016/0377-2217(91)90321-L)
- [13] Burke E, Kendall G and Soubeiga E. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*. 2003: 9 (6); 451-470. http://dx.doi.org/10.1023/B:HEUR.0000012446.94732_b6
- [14] Schaefer A and DiGaspero L. local search techniques for educational timetabling problems. In the Proceedings of International Symposium on Operations Research, Slovenia. 2001: 26-28.
- [15] Elmohamed S, Coddington P and Fox G. A Comparison of Annealing Techniques for Academic Course Scheduling. *Lecture Notes in Computer Science journal*, 1997: 1408; 92-114.
- [16] Kirkpatrick S, Gelatt C and Vecchi M. Optimization by simulated annealing. *Management Science Journal*, 1983: 220; 671-680.
- [17] Erben W, Keppler J. A Genetic Algorithm Solving a Weekly Course-Timetabling Problem. In the Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling. 1995: 21-32.
- [18] Muller T. Some Novel Approaches to Lecture Timetabling. 4th Workshop on Constraint Programming for Decision and Control (CPDC), Poland. 2002.
- [19] References Zervoudakis K and Stamatopoulos P. A generic object-oriented constraint-based model for university course timetabling. *Computer Science Journal*. 2001: 28-47.
- [20] References Russell S and Norvig P. *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2nd Edition. 2003: 137-160.
- [21] Daskalaki S, Birbas T and Housos E. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 2004:117-135. [http://dx.doi.org/10.1016/S0377-2217\(03\)00103-6](http://dx.doi.org/10.1016/S0377-2217(03)00103-6)

AUTHORS

Ibrahim Aljarah is with North Dakota State University, Fargo ND, USA.

Ayad Salhieh is with Jordan University of Science & Technology, Irbid, Jordan.

Hossam Faris is with University of Jordan, Amman, Jordan.

Received 22 November 2011. Published as resubmitted by the authors 28 February 2012.