MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 3, pp. 710–719 (2024)

A practical approach to college timetable scheduling

Thakur R. K.[1,2], Agrawal N. K.[2], Kumar P.[3]

[1] *Department of Mathematics, M.K. College, Darbhanga-846004, India*
[2] *Department of Mathematics, Lalit Narayan Mithila University, Darbhanga-846008, India*
[3] *Department of Mathematics and Scientific Computing, NIT Hamirpur, Himachal Pradesh-177005, India*

(Received 10 March 2023; Revised 16 September 2024; Accepted 18 September 2024)

This paper formulates the college timetable scheduling (CTTS) as a constraint satisfaction problem (CSP) in a manner that is easy to implement. Timetable scheduling is a process that is revised in every academic session and requires a lot of constraint checking, so it is inefficient to check the same type of constraints repeatedly to create a proper schedule manually. CSP is a natural choice for automating this process. The prototype presented in this paper is aimed to offer such assistance in scheduling classes for different courses running in an academic session keeping given different constraint checking like the limit check of the maximum number of lectures that can be assigned to a faculty, limit check of the maximum number of lectures that can be assigned to a course, detecting conflicts between courses scheduled in the same time slots, preventing overlapping assignments for faculty members in the same time slots. Here, first, the different aspects of the timetable scheduling problem are addressed, and then a technique is devised to help map the problem as a CSP.

Keywords: *college timetable scheduler; constraint satisfaction problem; multi-objective optimization; efficient solution.*

2010 MSC: 90B35 **DOI:** 10.23939/mmc2024.03.710

1. Introduction

Most timetable schedulers available for colleges are designed based on the curricula of universities and institutions outside of India. These schedulers often require significant modifications to work effectively for Indian universities. In this paper, a system is developed with consideration of the typical requirements and constraints faced by colleges in Indian universities. A college timetable for scheduling classes is based on a curriculum that includes various courses, which are further divided into different subjects to be taught in different semesters/years up to which the particular course runs. Every college at the university in India offers a number of different courses. These courses are run through one or more departments of the college. The curriculum of these courses is divided into different subjects in which some are interdepartmental (ID) in nature, i.e. these subjects can be opted by students of more than one department. These subjects are one of the main problems in scheduling as more than one department schedule is linked via these subjects. The more the number of these subjects the more complex the schedule is. Generic subjects, subsidiary subjects, and language subjects are examples of this nature. In the designed curriculum, there are limitations to the number of lectures that can be assigned to a particular subject in a week (termed as the workload of the subject) as well as there is a limitation to the number of lectures that can be assigned to a faculty member (workload of the faculty) depending upon their designation as per the rules and regulations of the University Grant Commission (UGC). While scheduling the lectures, it is essential to check whether the same time slot of a classroom is not provided for different subjects. Also, it is required to check whether two different lectures of the same faculty member are not assigned at the same time slot in a classroom. There are a lot of conditions like these to be met while scheduling a timetable for a college. These we termed as scheduling constraints.

The number of courses taught in a college decides the workload of a college in a particular year/semester. So, college timetable scheduling is, mainly, the process of scheduling subjects to be taught. To schedule a subject is to assign faculty members and time slots to that particular subject workload, which means that if a subject has 5 lectures a week workload, then scheduling that subject is to assign faculty members to those five lectures and 5 time slots to those 5 lectures in such a way that the scheduling constraints are satisfied. Therefore, assuming that an adequate number of classrooms and an adequate number of faculty members are available, scheduling a timetable for a college requires assigning:

1. Faculty members to a subject, and
2. Classroom time slots to that subject.

Keeping in view of the following:

1. No of lectures to be assigned to a subject must be equal to the number of lectures required for that subject, as per the curriculum.
2. There should be no collision between the timeslots of different subjects.

(It may happen that we will not be able to get a schedule that satisfies all the scheduling constraints due to a shortage of classrooms or scheduling problems due to constraints.)

Problem statement

For a College Time-Table Scheduler (CTTS), one wants to find an automated system that asks for some input and provide a feasible schedule as an output that satisfies all the required constraints.

The CTTS problem is formally described as follows: given a list of

1. Departments.
2. Courses with subjects associated.
3. Department wise faculty members details. Papers that are opted by more than one department i.e. ID subjects.
4. Number of classrooms, number of working days and number of lectures that a classroom can hold in a day.

The CTTS problem consists of scheduling lectures for each subject running in an academic session in such a way that

1. One subject can be taught by more than one faculty member (i.e., if a subject has a workload of 5 lectures, then it can be divided among more than one faculty member).
2. If a subject of one course is running, then at the same time slot another subject of that course cannot run, i.e., more than one subject of a course cannot run simultaneously.
3. Only one lecture is to be scheduled per day for a subject. If 2, 3, or 4 lectures need to be scheduled for a subject in a day, then they should be consecutive lectures.
4. A faculty member cannot be assigned to two different lectures at the same time slot in different classrooms.
5. The number of classes assigned to a faculty member should be equal to or less than the maximum number of lectures that can be assigned to that faculty member.
6. A lecture cannot be assigned to more than one faculty member, except for practical subjects or subjects of a similar nature.
7. The same time slot of a classroom on a particular working day cannot be assigned to two different subjects.

2. Literature review

Timetabling problem is diverse in nature. Educational timetabling is in general complex, highly constrained and multi-objective that interests researchers to come up with feasible solutions. Carter and Laporte [1] in its paper 'Recent Development in Practical Course Timetabling' reorganized various approaches from 1986 to 1996. They categorized the approaches into four categories. Valouxis and

Housos (2002) in their paper [2] uses constraint programming approach to come up with a solution and uses local search to enhance their obtained solution. In their work titled 'University Timetabling' Sanja Petrovic [3] pointed out that there is considerable overlap between different solution approaches to university timetabling problem. Several reviews (MirHassani and Habibi [4] (2011), Bettinelli [5] et al. (2015), Babaei [6] (2015), Vinod Kadam and Samir Yadav [7] – 2016) categorized major approaches to their solutions. Jat and Yang [8] (2011) proposed a two phased hybrid approach to solve university course timetabling problem. In its first phase Guided Search Genetic Algorithm was applied and in second phase a Tabu Search heuristic was used. Hakan Andersson [9] (2015) presented an automation for School Timetabling problem. In his thesis two meta heuristic algorithms with previous satisfying results, Simulated Annealing and Tabu Search, are implemented. In its study, Practices in Timetabling in Higher Education Institutions, 2017, Vrielink [10] provides a survey of different literatures on timetabling by discussing the differences and similarities in their theory and practice. In his work (2017), Md. A. Al-Jarrah [11], uses Genetic Algorithms (GAs) to build an automated course timetable system. Several other approaches, such as mathematical programming approach (Birbas [12] et al., 2009), hyper heuristics approaches (Soria-Alcaraz [13] et al., 2014), fuzzy multiple heuristic approach (Golabpour [14] et al., 2008), etc. are also used for their solution.

3. Design of CTTS

As the problem requires a lot of constraint checking, constraint satisfaction problem (CSP) is a natural choice to proceed with. Mathematically speaking, a CSP is a triplet (X, D, C) where X is a non-empty finite set of variables, D is the non-empty set of non-empty domains for the variables present in X and C is the set of constraints restricting the values that the variables can simultaneously take.

The CTTS require to schedule three different set of interlinked entities, the set of faculty members, the set of subjects of different courses and available classroom time slots. In totality the problem looks a little complex but the nature of the problem makes it easier to break into simpler fewer complex problems. Whatever the subject is taught in a college, it is taken care of by one particular department (means to say that a subject is not taught by faculty of two different departments). So, scheduling is less complex when viewed department wise. The major throne while scheduling is scheduling subjects of interdepartmental nature (ID subjects) i.e., subjects that are opted by more than one department. In our design, we create two layers of scheduling. In the first layer, a master timetable is created where time-slots of classrooms are allotted to all the subjects associated with every course. This process is carried out by a scheduler, we call it Master Time Table Schedular (MTTS). The MTTS then provide a schedule for each department in which time slots for different subjects are embedded in it. In the second layer, each department will assign a faculty member to the subjects associated with the courses running through it. This scheduling is done at Department level so we termed the scheduler as Department Time Table Schedular (DTTS). In short, MTTS will assign classroom time slots to subjects and DTTS will assign faculty to subjects.

3.1. Input for Scheduler

Before beginning the scheduling process, it is important to review the different types of input required for the scheduler to function effectively. A case study of HPS College, Madhepur, Bihar is used to illustrate the input parameters for the developed methodology. These inputs include details such as

— Departments
— Faculty members categorized by department
— Courses
— Department-wise subject details
— Classroom preferences
— Generic or subsidiary subject details,

as presented in Tables 1–6.

Table 1. Departments.

Department	Physics	Chemistry	Mathematics	Botany	Zoology	History	Economics	Political Science
Code	01	02	03	04	05	06	07	08
Department	Psychology	Sociology	Philosophy	English	Hindi	Sanskrit	Maithili	Urdu
Code	09	10	11	12	13	14	15	16

Table 2. Department-wise details of faculty members with codes.

Department	Name of Faculty	Designation	Allotted Lectures	Code
Physics	Phy Faculty 1	Associate Prof.	14	01_01_14
	Phy Faculty 2	Assistant Prof.	16	01_02_16
	Phy Faculty 3	Assistant Prof.	16	01_03_16
Chemistry	Chem Faculty1	Associate Prof.	14	02_01_14
	Chem Faculty2	Assistant Prof.	16	02_02_16
	Chem Faculty3	Assistant Prof.	16	02_02_16
Mathematics	Maths Faculty1	Associate Prof.	14	03_01_14
	Maths Faculty2	Assistant Prof.	16	03_02_16
	Maths Faculty3	Assistant Prof.	16	03_03_16
Botany	Botany Faculty1	Associate Prof.	14	04_01_14
	Botany Faculty2	Associate Prof.	14	04_02_14
	Botany Faculty3	Assistant Prof.	16	04_03_16
—	—	—	—	—

Coding Scheme faculty member:

$(DepartmentCode)\_(SerialNumberoffaculty)\_(MaximumlecturesallottedtotheFaculty)$.

Table 3. Course details.

Name of course	Course Code	Duration	Code of Departments Involved
B. A. (Honours)	01	3 years	06 to 15
B. Sc. (Honours)	02	3 years	01 to 05
I. A.	03	2 years	03, 06 to 15
I. Sc.	04	2 years	01 to 05, 11 to 15

Table 4. Subject Details.

Department	Course	Paper	Nature	Year	Allotted lectures	Code
Physics	B.Sc.(H)	Phy I	Honours Theory	Y1	5	01-02-01-HT-Y1-05
		Phy II	Honours Theory	Y1	5	01-02-02-HT-Y1-05
		Phy III	Honours Theory	Y2	5	01-02-03-HT-Y2-05
		Phy IV	Honours Theory	Y2	5	01-02-04-HT-Y2-05
		Phy V	Honours Theory	Y3	5	01-02-05-HT-Y3-05
		Phy VI	Honours Theory	Y3	5	01-02-06-HT-Y3-05
		Phy VII	Honours Theory	Y3	5	01-02-07-HT-Y3-05
		Phy Practical-1	Honours Practical	Y1	6	01-02-08-HP-Y1-06
		Phy Practical-2	Honours Practical	Y2	6	01-02-09-HP-Y2-06
		Phy Practical-3	Honours Practical	Y3	6	01-02-10-HP-Y3-06
		Phy Sub-1	Subsidiary Theory	Y1	5	01-02-11-ST-Y1-05
		Phy Sub-2	Subsidiary Theory	Y2	5	01-02-12-ST-Y2-05
		Phy Sub Practical-1	Subsidiary Practical	Y1	4	01-02-13-SP-Y1-04
		Phy Sub Practical-2	Subsidiary Practical	Y2	4	02-01-14-SP-Y2-04
	I.Sc.	Phy XI	General Theory	Y1	4	01-04-15-GT-Y1-04
		Phy XII	General Theory	Y2	4	01-04-16-GT-Y2-04
		Phy Practical-XI	General Practical	Y1	2	01-04-17-GP-Y1-02
		Phy Practical-XII	General Practical	Y2	2	01-04-18-GP-Y2-02
Chemistry	—	—	—	—	—	—

Subject Code Scheme:

$(DepartmentCode)\_(CourseCode)\_(SerialNumber)\_(Natureofpaper)\_(Maxlectureperweek)$, where max lecture per week is the allotted lectures.

Table 5. Classroom preference details for departments.

Class Room	Code	Specific Department that will use the classroom	% of Engagement
Room No-1	01	00	000
Room No-2	02	00	000
Physics Lab	03	01	100
—	—	—	—

Generic/subsidiary/language subjects are opted by students of more than one department, so these subjects are scheduled in such a manner that those students who opted these subjects has no other subjects scheduled, when these subjects are slotted i.e. If subject X is opted by students of departments D1 and D2 then while lectures of subject X are conducted, no other lectures of these students of the department D1 and D2 can be slotted. List of Interdepartmental subjects and departments involved.

Table 6. Generic Subject Details.

Subject Name	Subject Code	Departments Involved
Phy sub-1	01-02-01-ST-Y1-05	02, 03, 04, 05
Phy sub-2	01-02-02-ST-Y2-05	02, 03, 04, 05
—	—	—

It is to be noted that for a particular semester/year, the number of Generic/subsidiary/language subjects to be opted for a student of that semester/year of the same course must be equal, for example B.Sc.(H) Physics and B.Sc.(H) Mathematics both has to choose equal number of subsidiary subjects, equal number of language subjects.

Time slots table

If there are 7 lectures that can be assigned into a classroom in a day and the college runs for 6 days in a week, then one particular classroom can have $7 \cdot 6 = 42$ time slots that can be assigned by a scheduler. We can assign a unique code for each time slot as time slot code attached to room number code. So, room number one can have 42 time slots and the unique codes of each time slot for room number 1 are 01_01_01, 01_01_02, - - -, 01_01_07 for first day of a week lecture, 01_02_01, 01_02_02, - - -, 01_02_07 for second day of a week lectures, - - - and so on. In this way each time slot for each day of every classroom has a unique code.

Coding scheme for time slots:

$$(ClassroomCode)\_(DayNumber)\_(PeriodNumber).$$

Table 7. Time slots for room number whose code is 01.

	1st Period	2nd Period	3rd Period	4th Period	5th Period	6th Period	7th Period
Mon	01_01_01	01_01_02	01_01_03	01_01_04	01_01_05	01_01_06	01_01_07
Tue	01_02_01	01_02_02	01_02_03	01_02_04	01_02_05	01_02_06	01_02_07
Wed	01_03_01	01_03_02	01_03_03	01_03_04	01_03_05	01_03_06	01_03_07
Thu	01_04_01	01_04_02	01_04_03	01_04_04	01_04_05	01_04_06	01_04_07
Fri	01_05_01	01_05_02	01_05_03	01_05_04	01_05_05	01_05_06	01_05_07
Sat	01_06_01	01_06_02	01_06_03	01_06_04	01_06_05	01_06_06	01_06_07

3.2. Scheduling strategies

— Feasibility Check - If (No. of lectures per week that can be assigned to a classroom)×(No.of class room)⩾(sum of workload of each department) then there may exists a solution, provided that the problem is not over constraint. Here by workload of a department we mean the number of lectures that is required by the department to run all its courses. If scheduling is not achieved then either reduce the workload or increase the number of class rooms or use the same class room twice, one for morning schedule and another for evening schedule will help to get to a solution.

— Creation of Master Time Table(MTT) - With the knowledge of workload of each department, if we partition the total available lecture time-slots in a week to different departments then with the provided time-slots in such a way that the time slots for each subject of each department is assigned then the only requirement for a solution is to assign faculty to these time slots. This bifurcated schedule containing time-slots of each subject for different departments is termed Master Time Table (MTT). Since generic/subsidiary/language subjects are opted by students of more than one department, it is a major problem in scheduling as a lot of department schedules are connected by these subjects. The MTTS, first schedule ID subjects and then go for scheduling non-ID subjects.

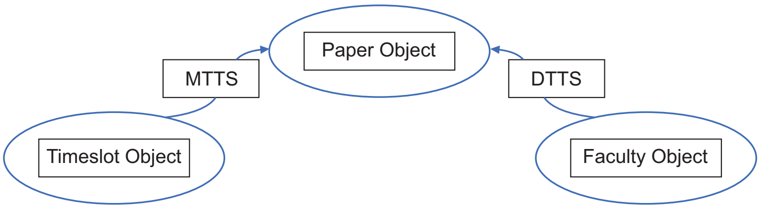


Fig. 1.

Four types of constraint checking are required while preparing MTT.

1. Room preference constraint.
2. More than one faculty can be assigned to practical subjects.
3. A faculty cannot be assigned to two different subjects at the same time.
4. Limit check to the maximum number of assignments that can be made to a faculty member.

The output of MTT scheduler is a list (see Table 8).

Table 8. MTT Output Table.

Department code	Subject code	Fixed Time slots for Subject codes				
01	02-01-01-ST-Y1-05	Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5
		Timeslot1	Timeslot2	Timeslot3	Timeslot4	Timeslot5
	—	—	—	—	—	—
—	—	—	—	—	—	—

Using the above list each department retrieve their individual schedule and use it to create their own timetable by assigning faculty to the allotted time slots to the subjects concerning their department with the help of departmental timetable (DTT) scheduler.

3.3. Creation of departmental timetable

A Departmental Time-Table (DTT) scheduler will solve the last stage of the problem by assigning faculty to the assigned time slots of the subjects concerning a department once at a time. Since the time slots are portioned among different departments by MTTS so there is no collision among time slots of different departments. A concurrent approach can be made to get departmental timetables by using constraints related to faculty of the concerned department.

The following types of constraints are required to be taken care of:

1. Faculty preferences
2. More than one faculty can be assigned to practical subjects.
3. A faculty cannot be assigned to two different subjects at the same time.
4. Limit check to the maximum number of assignments that can be made to a faculty member.

4. The scheduling model

4.1. Variables

There are three types of variables: Subject, Faculty and Timeslots.

1. Subject object.

Number of subject object = number of subjects to be taught in current session.

Subject class description is given in Table 9.

Table 9. Subject.

Integer assignmentLevel=0;	Used to check whether the object is assigned or not, if assignmentLevel < workload of paper then assignment of paper is incomplete
Integer workload;	Store number of lecture to be allotted to the paper
String timeslotCode[workload];	an array of string, use to store timeslots for the associated lectures of the paper
String facultyCode[workload];	an array of string, use to store faculty codes for the associated lectures of the paper corresponding to assigned timeslots codes
Set, get functions for members of the class	
constructor of the class	

2. Faculty object.
Number of faculty object = number of faculty in total for the session.
Faculty class description.

Table 10. Faculty class.

Integer assignmentLevel=0;	Used to check whether the object is assigned or not, if assignmentLevel <workload of faculty then assignment of the faculty is incomplete.
Integer workload;	Store number of lecture to be allotted to the faculty.
String timeslotCode[workload];	an array of string, use to store timeslots for the associated lectures of the paper
String paperCode[workload];	an array of string, use to store papercodes for the associated lectures of the paper corresponding to assigned timeslots codes
set, get functions	
constructor of the class	

3. Timeslot object.
Number of timeslot object = (no. of classroom)×(no. of lecture per week)×(no of working days in a week).
Timeslots class description is given in Table 11.

Table 11. Timeslot class.

Integer assign=0;	Used to check whether the object is assigned or not
String paperCode;	Store paperCode to which it is allotted
set, get functions	
constructor of the class	

4.2. Domains for the variables

Domain of Subject objects is a set consisting of the subjects (timeslot code, faculty code).

Domain for Faculty object is a set consisting of the pairs (timeslot code, paper code)of faculty code.

The Domain for Timeslots is the set of timeslot codes.

4.3. Constraints for MTTS: (MTTS assigns timeslotCodes to Subject objects)

Number of variables = number of Subject objects.

We create Subject object with its name as paperCodes means to say that the object name of a subject is its paperCode. For example, Subject class object names are 01-02-01-HT-Y1-05, 01-02-02-HT-Y1-05, - - - etc. We then partition the set of Subject objects into two different sets of ID Subjects and Non-ID Subjects. This can be done by checking the name of Subject objects which contains the nature of the subject for example, the Subject object 01-02-01-HT-Y1-05 is Honours Theory (HT) subject so it is in the set of Non-ID Subjects. Assignment for ID Subjects are done first, then assignment of Non-ID Subjects are to be made.

Two set of time slots are to be managed, named as Assigned Time slots and Unassigned Time slot. If an unassigned time slot is slotted to a subject, then that time slot object is to be removed from the set of Unassigned Times lots to the set of Assigned Time slots.

Assignment is to be made from unassigned time slots.

Constraint_1.

First, the assignment of ID subjects that are opted by maximum departments to be done. For this, a table of ID Subjects with decreasing number of opted departments is formed (Lets name the table, Department Opted (DO) Table, this table can easily be obtained by Input Table 6).

Constraint_2.

Assignment for any subject is made up to its workload.

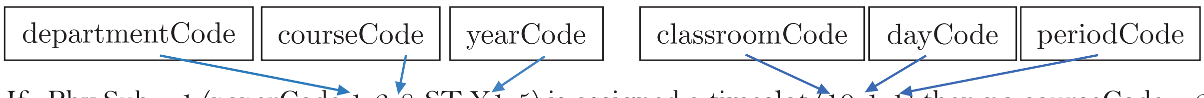
Constraint_3.

Using Input Table 5, Assignment is to be made from the set of Unassigned Time slots from preferred classroom time slots such that

— Practical Subject assignments must be consecutive and once in a week.
— Non-Practical Subject are to assigned once in a day.

Constraint_4.

No two different subjects of the same course and of the same year is to be slotted at the same time at different classroom. For example,



If, Phy Sub – 1 (paperCode 1-2-8-ST-Y1-5) is assigned a timeslot (10-1-1) then no courseCode – 2 papers of opted departments (the departments that opted this paper) of first year (Y1) be assigned in any classroom with dayCode – 1 and periodCode – 2. (i.e., timeslotCodes of the type *-1-1 are prohibited for courseCode – 2 yearCode – 1 papers from opted departments mentioned in Input Table – 6)

Note:

1. Constraint checking is made from assigned time slots only.
2. Set of assigned time slots is further subdivided into sets of time slots with same courseCode and sameYear, i.e., different set named as courseCode-yearCode are to be maintained which contains time slots assigned to subject related to that courseCode-yearCode combination. Whenever an assignment is to made for a subject of the combination courseCode-yearCode then constraint checking for the assignment are to be made from that set associated with its coursecode-yearCode combination set.

Once assignment of all ID Subjects is made then assignment of Non-ID Subjects are to be made keeping in view of the same four types of constraints as above.

If all the subject assignments are done successfully then MTTS work is over and it is the DTTS which takes over to the proceedings. If not, then the problem is over constraint. If the number of ID Subjects are too high then increasing the number of classes per day could help. If the number of Non-ID Subjects are too high then increasing the number of class room could help.

4.4. Constraints for DTTS

If MTTS runs successfully then assignment of time slots to all the subjects is completed. What it remains is to assign faculty to these subjects. Since all departments have knowledge of the subjects associated with it and these subject object have timeslot attached to it, so the DTTS will assign faculty code section of subject objects of different departments one by one. Constructing a table named Preference Table of subjects that associate facultyCodes of those faculty who prefer that subject to teach. Arrange that table in ascending order to the number of choices to subjects. Assignment of the subject which has a smaller number of choices is to made first.

DTTS takes subjects associated with one department and faculty associated with that department then assigns facultyCodes to Subject objects one by one keeping in view of the following two types of constraints:

Contaraint_1: (Optional constraints for graduate teachers).

Assignment of faculty to Subject objects is to made from preference table.

Constraint_2.

A faculty cannot be assigned to two different class room at the same time i.e., in Faculty object timeslot array, if $a_1 - b_1 - c_1$ and $a_2 - b_2 - c_2$ are two entries then $b_1 \neq b_2$ and $c_1 \neq c_2$.

Note:

1. Choice of subjects by faculty members must be evenly distributed means to say that all subjects are to opted by about same number of faculty for feasibility for first constraint satisfaction.
2. Once a valid assignment is obtained then both subject objects and faculty objects entries must be updated accordingly.

Upon successful completion of DTTS, a required schedule is obtained.

5. Conclusion

Most of the scheduling routine is usually done by MTTS. If MTTS runs successfully, we will obtain a schedule by either successful completion of DTTS or upon unsuccessful completion of DTTS by relaxing or modifying constraints for DTTS.

The major advantage of the scheduler lies in its flexibility: Any college can use this tool.

Another advantage of this scheduler is that a change in curriculum can easily be incorporated by minor modifications in the scheduler.

The scheduler can provide schedules for each subject, each classroom, each faculty separately.

The issue affecting any CSP's performance is the number of constraints checking which further reduces the performance of any algorithm if the domain set of associated variables is large. To overcome the issue of a large set of variables, the domain set is subdivided and an additional constraint has to be introduced (MTTS-Note-2).

The tailoring process is implemented in any object-oriented programming language (e.g., C++, java, python).

In future, performance of the scheduler may be enhanced by reducing the same problem as the distributed constraint satisfaction problem.

[1] Carter M. W., Laporte G. Recent developments in practical course timetabling. Practice and Theory of Automated Timetabling II. 3–19 (1997).

[2] Valouxis C., Housos E. Constraint programming approach for school timetabling. Computers & Operations Research. **30** (10), 1555–1572 (2003).

[3] Petrovic S., Burke E. K. University timetabling. Handbook of scheduling (2004).

[4] MirHassani S. A., Habibi F. Solution approaches to the course timetabling problem. Artificial Intelligence Review. **39** (2), 133–149 (2013).

[5] Bettinelli A., Cacchiani V., Roberti R., Toth P. An overview of curriculum-based course timetabling. TOP. **23** (2), 313–349 (2015).

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

[7] Kadam V., Yadav S. Academic timetable scheduling: revisited. International Journal of Research in Science & Engineering. 417–423 (2016).

[8] Jat S. N., Yang S. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. Journal of Scheduling. **14** (6), 617–637 (2011).

[9] Andersson H. School timetabling in theory and practice a comparative study of simulated annealing and tabu search. UMEÅ Universitet (2015).

[10] Oude Vrielink R. A., Jansen E. A., Hans E. W., van Hillegersberg J. Practices in timetabling in higher education institutions: a systematic review. Annals of Operations Research. **275** (1), 145–160 (2019).

[11] Al-Jarrah M. A., Al-Sawalqah A. A., Al-Hamdan S. F. Developing acourse timetable system for academic departments using genetic algorithm. Jordanian Journal of Computers and Information Technology (JJCIT). **3** (1), 25–36 (2017).

[12] Birbas T., Daskalaki S., Housos E. School timetabling for quality student and teacher schedules. Journal of Scheduling. s**12** (2), 177–197 (2009).

[13] Soria-Alcaraz J. A., Ochoa G., Swan J., Carpio M., Puga H., Burke E. K. Effective learning hyperheuristics for the course timetabling problem. European Journal of Operational Research. **238** (1), 77–86 (2014).

[14] Golabpour A., Shirazi H. M., Farahi A., Kootiani A. Z. M., Beigi H. A Fuzzy Solution Based on Memetic Algorithms for Timetabling. 2008 International Conference on MultiMedia and Information Technology, Three Gorges. 108–110 (2008).

Практичний підхід до складання розкладу коледжу

Такур Р. К.[1,2], Агравал Н. К.[2], Кумар П.[3]

[1] *Кафедра математики, М.К. Коледж, Дарбханга-846004, Індія*
[2] *Факультет математики, Університет Лаліт Нараян Мітхіла, Дарбханга-846008, Індія*
[3] *Кафедра математики та наукових обчислень, NIT Хамірпур, Хімачал Прадеш-177005, Індія*

У цій статті сформульовано планування розкладу коледжу (CTTS) як проблему задоволення обмежень (CSP) способом, який легко реалізувати. Планування розкладу — це процес, який повторюється кожної академічної сесії та вимагає перевірки численних обмежень. Ми вважаємо, що неефективно проводити багатократні перевірки тих самих обмежень вручну. CSP є природним вибором для автоматизації цього процесу. Прототип планувальника розкладу, поданий у цій статті, має на меті запропонувати таку допомогу в плануванні занять для різних курсів, що проводяться під час академічної сесії, зважаючи на різні перевірки обмежень, таких як: перевірка обмеження на максимальну кількість лекцій, які можуть бути призначені факультету; перевірка обмежень максимальної кількості лекцій, які можуть бути призначені для курсу; перевірка конфліктів між курсами, запланованими для однакових часових проміжків; запобігання збігу аудиторій, призначених для викладачів у ті самі часові проміжки, тощо. У статті спочатку розглядаються різні аспекти проблеми планування розкладу, а потім розробляється техніка, яка покликана допомогти відобразити проблему як CSP.

Ключові слова: *планувальник розкладу в коледжі; проблема задоволення обмежень; багатоцільова оптимізація; ефективний розв'язок.*