International Journal of Combinatorial Optimization Problems and Informatics, 11(2), May-Aug 2020, 46-60. ISSN: 2007-1558.

www.editada.org

Case study: Three stages in the planning of postgraduate examinations through binary integer programming

José Israel Hernández-Vázquez[1], Salvador Hernández-González[2], María del Rosario Baltazar-Flores[3], José Alfredo Jiménez-García[2], José Omar Hernández-Vázquez[1], Moisés Tapia-Esquivias[2].
[1]Tecnológico Nacional de México en Celaya, Departamento de Doctorado en Ciencias de la Ingeniería.
[2]Tecnológico Nacional de México en Celaya, Departamento de Ingeniería Industrial.
[3]Tecnológico Nacional de México en León, División de Estudios de Posgrado e Investigación.
E-mail: d1703004@itcelaya.edu.mx, salvador.hernandez@itcelaya.edu.mx, rosario.baltazar@itleon.edu.mx, alfredo.jimenez@itcelaya.edu.mx, d1703005@itcelaya.edu.mx, moises.tapia@itcelaya.edu.mx

Abstract. The scheduling of examination timetables is a problem that arises in educational institutions such as universities, high schools and junior high schools. Mathematical programming models are used to solve this administrative problem. This is known as an NP-complete problem from the perspective of computational complexity, because of the large number of possible timetable combinations. This article presents a new strategy for defining the scheduling of examinations by stages, by decomposing the original problem, using binary variables, into three mathematical models. The assignment considers students, time slots, classrooms and examiners. We have taken the Department of Postgraduate Studies of the Tecnológico Nacional de México in Celaya as a case study. The strategy generated a significant reduction in the number of binary variables, Making it possible for the exact technique of branch and bound to reach efficient times in the search for an optimum solution at each stage.
Keywords: Examination timetabling, Integer programming binary, Combinatory optimization, NP-Complete.

Article Info
Received Sep 11, 2018
Accepted Sep 11, 2019

1 Introduction

The definition of advance timetables is a problem that can be found in different spheres but particularly in educational institutions. Their complexity lies in the huge number of possible timetable combinations, which means that choosing the best or a good solution in a reasonable time that will meet the needs of the institution is a task that can take up a massive amount of time, even in the case of problems that only involve a handful of courses, classrooms and professors.

The existing literature on scheduling examination timetables gives different solution approaches that use mathematical modelling; particularly the use of various methods to search for good solutions in short times or the use of exact techniques in the search for an optimum solution.

Nowadays the definition of school timetabling is known as an NP-complete problem [1, 2], owing to a large number of combinations present, which means that there is a drastic increase in the time spent on the search for a solution. So we need a treatment that will shorten the feasible solution space.

The main contribution of this article is a strategy for bounding the search space, by defining the timetable by stages. Although the literature has reported similar procedures using mixed-integer programming, it is worth mentioning that unlike these, the strategy we implemented uses binary integer programming.

The strategy consists of decomposing the original problem into three mathematical models, employing binary variables with two indexes, subsets and a heuristic method that defines the coefficients of the first model's objective function.

The rest of this document is organized as follows: Section 1.1 refers to relevant articles on defining examination timetables to be found in the literature, combined with a brief explanation of Examination Timetabling in section 1.2. Section 2 describes the case study problem. Section 3 gives a step-by-step breakdown of the methodology implemented in the search for a solution and presents the mathematical models developed in each stage. Section 4 details the characteristics taken into account for the experimentation and the results obtained; then section 4.1 establishes the computational results for the assessment of the mathematical models in large-scale problems. Lastly, section 5 gives the conclusions of our research.

1.1. Review of the literature

Historically, the literature has provided different approaches to solving the Examination Timetabling. A significant number of authors have opted for getting good solutions in short times, that meet the needs of educational institutions using different strategies, including the use of local search methods [3, 4, 5]; metaheuristic techniques: tabu search [6, 7, 8], simulated annealing [9] and evolutionary algorithms [10, 11, 12]; multi-objective methods [10, 13, 14, 15]; the use of hyper-heuristics [16, 17, 18, 19, 20, 21]; the GRASP method [22], graph coloring [23], the GDA (Great Deluge Algorithms) [24] and CG (Column Generation) [25].

In another line of research, the authors have opted for the use of integer programming and exact techniques to search for an optimum solution. One of the most outstanding proponents of this technique is MirHassani [26], who developed a mathematical model aimed at minimizing the conflicts involved with scheduling two examinations at the same time, on the same day or on consecutive days. For their part, Al-Yakoob et al. [27] employed mixed integer programming to define examination timetables in the university of Kuwait, using subsets in the mathematical modelling over two stages: the first stage considers the assignment of the examinations to a class time slot, then, in the second stage a person is assigned to be in charge of applying the examination. Further on McCollum et al. [28] develop a model based on the examination timetabling needs of European, Australian and American universities.

More recently, Cataldo et al. [29], propose a solution approach based on three mixed mathematical programming models that are solved in sequence. The first model assigns time slots and types of the classroom to the groups of courses being assessed. The second model assigns time slots and types of the classroom to individual courses, as per the solution from the first model. Finally, the third model assigns specific classrooms to the examinations for each course. The mathematical models were, for their validation, applied to a series of real cases from the Universidad Diego Portales in Chile.

Moreover, Kasm et al. [30] propose two alternatives for generating solutions in examination timetabling at universities: the first consists of a mathematical model using binary variables to solve small instances, while the second contemplates the use of a heuristic that enables us to generate high-quality solutions in large instances. Then we finally compare them with each other to validate the efficiency of the heuristic.

1.2. Characteristics of the Examination Timetabling

Examination Timetabling problems can be defined as the assignment of a set of examinations, within a limited number of time slots, considering a group of classrooms of limited capacity and a set of constraints, as mentioned in [31].

These problems are subject to a large variety of constraints that are generally divided into two categories: "hard" and "soft". The hard constraints are strictly applied and must not be violated, whereas the soft constraints are desirable but not absolutely essential.

Examples of hard constraints:

- Only one group of students and one professor can attend one classroom in each time slot.
- A professor or student cannot attend two examinations at the same time.
- The classrooms' capacity must be in proportion to the number of students being assessed.

Examples of soft constraints:

- The professor has the option of suggesting priorities in certain time slots for their assessments.
- Examinations must be scheduled in such a way as to minimize the teachers' empty time slots.

2. Description of the problem (case study)

For the generation of examination timetabling, we took the Tecnológico Nacional de México in Celaya, Guanajuato as a case study. This educational institution offers eleven postgraduate courses in different areas of engineering with a variety of specialties where the students can develop their research.

At the end of every semester, the individual students from the different postgraduate courses present the progress of their research projects to a jury of professors-researchers who question and evaluate their performance.

The educational institution takes an average time of one week every semester and one person per engineering area for timetabling the final exams and its criterion is the trial-and-error allocation of resources, seeking to meet each postgraduate student's needs.

The following constraints are considered when drawing up the timetable for the assessments:

1. There are 5 available time slots per day in each classroom.
2. Certain days of the week are set aside in each postgraduate course for project assessment.
3. We have a limited number of classrooms per postgraduate course, which restricts the number of available time slots to be assigned to students for their assessment.
4. Students in the same specialty should preferably be assigned to different consecutive time slots, so that they can be assessed by professors of subjects that are compatible with the specialty.
5. Priority shall be given in the assessments to the first time slots of each day.
6. The least possible number of days should be used for the assessment of students.
7. Each classroom should preferably be assigned to students in the same specialty.
8. No two students can be assigned to a particular classroom on the same day and in the same time slot.
9. Examiners will be assigned to the students according to the compatibility of the research project with the professor's area of knowledge. There are three categories: 1- High compatibility, 2- Medium compatibility and 3- Low compatibility.
10. Students should be assessed by a certain number of professors.
11. Each professor has a minimum and maximum number of students to assess.
12. Professors cannot act as assessors to their own students under assessment.
13. A professor can only assess one student in each time slot.
14. Students under assessment must be assigned to the time slots that each professor has available.

3. Method

The number of combinations of a mathematical model with binary variables is 2^n, where n is the number of variables. To bound the search space, we propose the strategy of solving the problem by stages, based on the papers presented by Al-Yakoob et al. [27] and Cataldo et al. [29], which will enable us to lower the number of binary variables by setting up the examination timetabling in three mathematical models.

Unlike the studies presented in [27, 29] where mixed integer programming is used, the research described in this article only uses binary variables with two indexes in each stage, in addition to using a heuristic for the definition of coefficients of the objective function in stage 1.

Examination timetabling is started by ordering the students according to specialty, followed by setting the timetable up sequentially in three stages (3 mathematical models). The first consists of assigning a time slot to each student, then in the second, the assessment classroom is assigned and lastly, during the third, the examiner(s) are designated (see Fig. 1).

Table 1 makes a comparison between using a mathematical model to set up an examination timetable, as in the studies presented in [26, 28, 30] that use integer programming and exact techniques to search for a solution, and the proposal we are making in this article for the use of three mathematical models, taking into account 50 students, 5 time slots, 5 classrooms and 16 professors. As can be seen, when the model is decomposed into three stages, the number of variables is considerably reduced by 93.5% and it is only necessary to use two indexes in each variable.

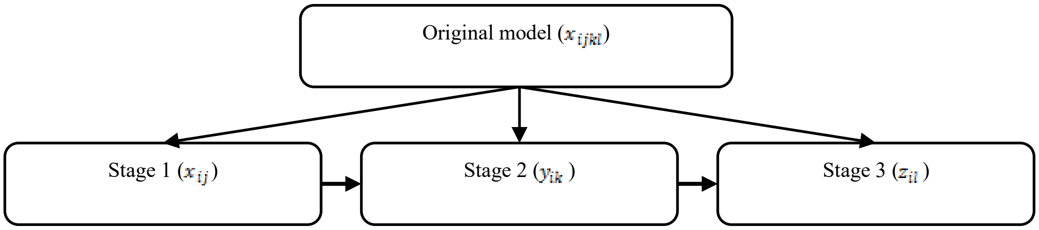

Fig. 1 Decomposition of the original problem into three stages.

Table 1. Comparison of the number of binary variables used, when defining the timetable using 1 and 3 mathematical models

1 Mathematical model		3 Mathematical models					
Model 1 x_{ijkl}		Model 1 x_{ij}		Model 2 y_{ik}		Model 3 z_{il}	
Binary variable 4-indexes		Binary variable 2-indexes		Binary variable 2-indexes		Binary variable 2-indexes	
i Student	50	*i* Student	50	*i* Student	50	*i* Student	50
j Time slot	5	*j* Time slot	5	*k* Classroom	5	*l* Professor	16
k Classroom	5	Variables (*i* x *j*)	250	Variables (*i* x *k*)	250	Variables (*i* x *l*)	800
l Professor	16						
Total of binary variables (*i* x *j* x *k* x *l*)	20,000	Total of binary variables	1,300				

Subsets are also used in each of the mathematical models to facilitate mathematical modelling and reduce the interaction between the variables, as done by Al-Yakoob et al. [27] and Cataldo et al. [29]. Moreover, coefficients are intentionally used in the objective function to speed up the search in certain binary variables.

3.1. Notation employed in the development of the mathematical models

Sets

I Students to be assigned $I = \{Student_1, \mathrm{Student}_2, ..., \mathrm{Student}_{|I|}\}$

J Available time slots for sitting the examination per day $J = \{T_1, \mathrm{T}_2, ..., \mathrm{T}_{|J|}\}$

K Classrooms available for the assignment of students $K = \{C_1, \mathrm{C}_2, ..., C_{|K|}\}$

L Professors available for the assessment $L = \{P_1, P_2, ..., P_{|L|}\}$.

Indexes

i Student $i \in I$
j Time slot $j \in J$
k Classroom $k \in K$
l Professor $l \in L$

Subsets

$I_m \subset I$ {Students in the same specialty m }.
$I_n \subset I$ {Students assigned to the different time slots $n = \{1, 2, 3, ..., 5\}$

Coefficients of the objective function

a_{ij} = Compatibility of the student with the time slot.

a_{ik} = Compatibility of the student's specialty with each classroom.

a_{il} = Compatibility of the professor with the topic being assessed for each student.

Parameters

$Available\ clasrooms =$ Number of available classrooms for the assignment.
$STUDTIME =$ Maximum number of students per time slot in each classroom.
$PMIN_i =$ Minimum number of professors per student *i*.
$PMAX_i =$ Maximum number of professors per student *i*.
$SMIN_l =$ Minimum number of students per professor *l*.
$SMAX_l =$ Maximum number of students per professor *l*.
$DAY_l =$ Monday and Tuesday.
$TIMEP_l =$ Time slots when assessments can be assigned to each professor *l*.

Binary variable used in stage I: Student-Time slot Assignment
x_{ij} 1 Student *i* is assigned to time slot *j*.
0 Student *i* is not assigned to time slot *j*.

Binary variable used in stage II: Student-Classroom Assignment
y_{ik} 1 Student *i* is assigned to classroom *k*.
0 Student *i* is not assigned to classroom *k*.

Binary variable used in stage III: Student-Professor Assignment
Z_{il} 1 Student *i* is assigned to professor *l*.
0 Student *i* is not assigned to professor *l*.

3.2. Stage I: Student-Time slot Assignment

During this stage, students are assigned to time slots by considering the first six constraints mentioned in the description of the problem (section 2).

The construction of the mathematical model is based on subsets I_m made up by students in the same specialty. One essential part of the model's development is the intentional setting of coefficients a_{ij} in the objective function in order to:

- Assign students in the same specialty to consecutive time slots, so as to give the professors corresponding to the specialty the opportunity to assess them.
- Lower the number of examination days, by assigning the highest possible number of students during the first assessment days.
- Preferably assign the highest number of students to the first time slots of each day during the assessments.

Heuristic method for defining coefficients in the objective function a_{ij}

Before establishing the coefficients in the objective function, the minimum number of classrooms required for the preparation of the timetable is calculated by using equation (1), rounding the value obtained up to the higher integer:

$$Minimum\ number\ of\ classrooms\ needed\ = Roundup\left[\frac{Number\ of\ students}{Time\ slots * Exam\ days}\right] \tag{1}$$

Once the minimum number of classrooms required has been defined, the number of physically available classrooms is set in order to define the timetable. Obviously, this number must be equal to or higher than the result of the calculation obtained using equation (1) to generate a feasible solution, as shown in inequality (2):

$$Available\ classroom\ \geq Minimum\ number\ of\ classrooms\ classrooms\ required \tag{2}$$

Then the maximum number of students per time slot in each classroom is determined using equation (3), rounding the value obtained up to the higher integer. This number also defines the days needed for the examination timetabling.

$$STUDTIME\ = Roundup\ \left[\frac{Number\ of\ students}{Time\ slots * Available\ classrooms}\right] \tag{3}$$

Blocks are used to set coefficients a_{ij}. Each one contains students in the same specialty that are preferably assigned to the same day, in the different available time slots. The form of Table 2 is used to record the coefficients a_{ij} of each block.

The coefficients of the objective function a_{ij} are set based on a value X_1 that is proposed by the person in charge of scheduling the examination timetables, following the sequence in the flow diagram shown in Fig. (2), this value must be higher than the number of students that are contemplated per specialty. Once they have been defined, they are substituted in the mathematical model of the first stage.

Table 2. Form for filling in coefficients of objective function a_{ij} of block *w*.

Objective function coefficients								
Block	Student	Time slots						
		T1	T2	T3	T4	T5	…	Tn
w	1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	…	C_{1n}
	2	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	…	C_{2n}
	3	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	…	C_{3n}
	4	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	…	C_{4n}
	5	C_{51}	C_{52}	C_{53}	C_{54}	C_{55}	…	C_{5n}
	.	.	.	.	.	.	…	.
	m	C_{m1}	C_{m2}	C_{m3}	C_{m4}	C_{m5}	…	C_{mn}

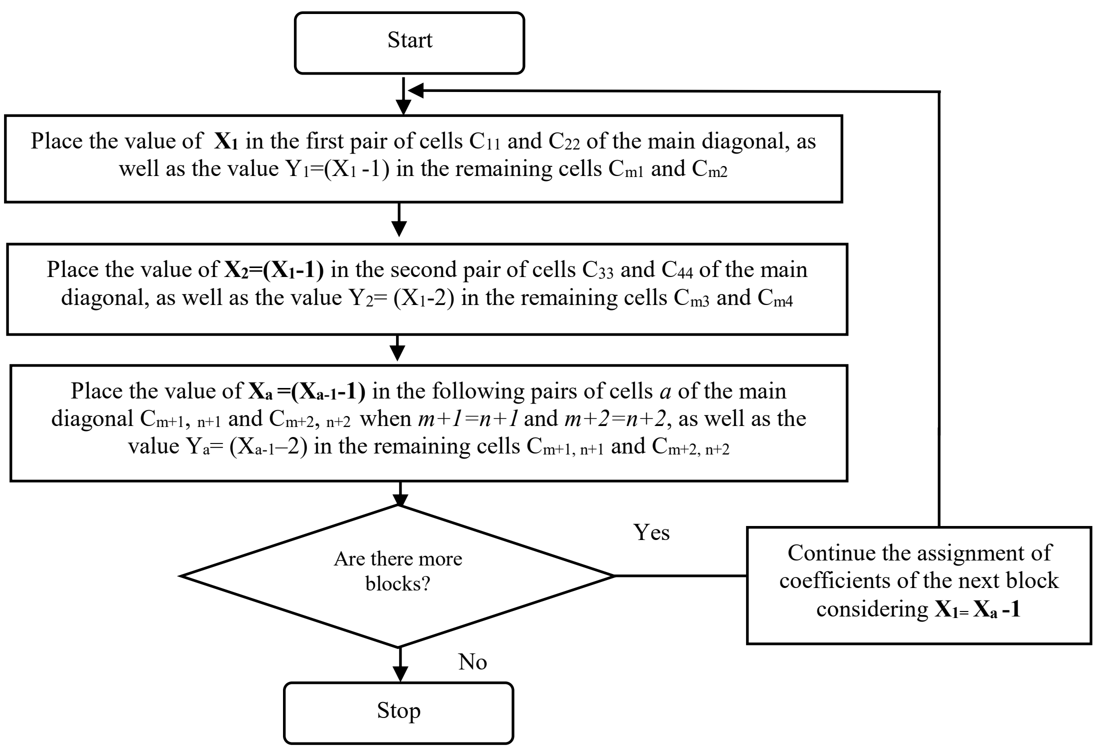

Fig. 2. Flow diagram for filling coefficients in objective function a_{ij}.

Mathematical model 1: Student-Time slot Assignment

$$Z_{\mathrm{Max}} = \sum_{i \in I_m} \sum_{j \in J} a_{ij} x_{ij} \tag{4}$$

Subject to: (5)

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I_m$$

$$\sum_{i \in I_m} x_{ij} \leq (Available\ clasrooms)(STUDTIME) \quad \forall j \in J \tag{6}$$

$$x_{ij} \in \{0,1\} \tag{7}$$

Objective function (4) seeks to maximize the assignment of students in the same specialty to consecutive time slots, in the first days and times for assessment.
As for the constraints:

- (5) Makes the students be assigned to no more than one-time slot for their assessment.
- (6) Makes it impossible to exceed the number of students assigned per time slot, of the classrooms available on the assessment days.

The result of this first stage defines the assignment of students to each subset I_n to be used in stages II and III.

3.3. Stage II: Student-Classroom Assignment

In this second stage, students are assigned to a classroom, considering constraints 7 and 8 mentioned in the description of the problem (section 2).

The construction of the mathematical model is based on subsets consisting of students assigned to the same time slot I_n.

During this stage of the solution process, we seek to accommodate the highest number of students in the classroom as per their area of specialty; the coefficients corresponding to objective function a_{ik} weight the degree of the student's compatibility with the classroom to be scheduled. The value of 5 is assigned to students with topics that are compatible with the classroom's specialty or else the value of 4 is assigned to students with partially related topics. The grade is set by the coordinator of postgraduate studies and the person in charge of scheduling the examination timetables. Once coefficients a_{ik} have been defined, they are substituted into the mathematical model of the second stage.

Mathematical model 2: Student-Classroom Assignment

$$Z_{\mathrm{Max}} = \sum_{i \in I_n} \sum_{k \in K} a_{ik} y_{ik} \tag{8}$$

Subject to: (9)

$$\sum_{k \in K} y_{ik} \leq 1 \quad \forall i \in I_n$$

$$\sum_{i \in I_n} y_{ik} \leq STUDTIME \quad \forall k \in K \tag{10}$$

$$y_{ik} \in \{0,1\} \tag{11}$$

The objective function (8) seeks to maximize the assignment of students to the classroom of their specialty.
For their part, the constraints:

- (9) Make each student be assigned to no more than one classroom.
- (10) Make it impossible to exceed the maximum number of students per time slot in each classroom.

3.4. Stage III: Student-Professor Assignment

In this last stage, the students are assigned to one or more examiners, considering the group of constraints 9-14 that is mentioned in the description of the problem (section 2). The construction of the mathematical model is based on the same stage II subsets.

The model developed in this third stage seeks to maximize the professor's compatibility with each student's topic under assessment; the process for defining the coefficients of objective function a_{il} follows a similar process to the one presented in the previous stage, establishing a rating scale. The value of 5 corresponds to professors that have high compatibility with the topic under assessment; the value of 4 includes professors with medium compatibility; the value of 3 considers professors with low compatibility; and, lastly, 1 when the professor is not compatible with the topic or is acting as an adviser to the student under assessment. Once coefficients a_{il} have been defined, they are substituted in the third stage's mathematical model as shown below:

Mathematical model 3: Student- Professor Assignment

$$Z_{\text{Max}} = \sum_{i \in I_n} \sum_{l \in L} a_{il} z_{il} \tag{12}$$

Subject to: (13)

$$\sum_{l \in L} z_{il} \geq PMIN_i \quad \forall i \in I_n$$

$$\sum_{l \in L} z_{il} \leq PMAX_i \quad \forall i \in I_n \tag{14}$$

$$\sum_{i \in I_n} z_{il} \geq SMIN_l \quad \forall l \in L \tag{15}$$

$$\sum_{i \in I_n} z_{il} \leq SMAX_l \quad \forall l \in L \tag{16}$$

$$\sum_{i \in I_n} z_{il} \leq DAY_l \quad \forall l \in L \tag{17}$$

$$\sum_{i \in I_n} z_{il} \leq TIMEP_l \quad \forall l \in L \tag{18}$$

$$z_{il} \in \{0,1\} \tag{19}$$

Objective function (12) seeks to maximize the compatibility of the professors with the different topics under assessment for each student.

For their part, the constraints:

- (13) Forces the assignment of the minimum number of examiners per student.
- (14) Makes it impossible to exceed the maximum number of examiners per student.
- (15) Forces the assignment of the minimum number of students being assessed per professor.
- (16) Makes it impossible to exceed the maximum number of students being assessed per professor.
- (17) Forces the assignment of no more than one student to each professor per time slot, in each one of the assessment days.
- (18) Forces the assignment of students to the professors' available time slots.

4. Experiments and results in the study case

In the area of postgraduate studies of the Tecnológico Nacional de México in Celaya, each department organizes its own examination timetables. To demonstrate the effectiveness of the mathematical models developed in each stage, we chose as a case study the January-June 2019 academic cycle of the industrial engineering postgraduate department. This has the highest enrolment of students in the institution, which is why the efficient use of resources in the timetabling of examinations is very important.

Nine experiments were performed to validate the mathematical modelling, considering a maximum of 50 students, 5-time slots (T1 18:00-18:30, T2 18:30-19:00, T3 19:00-19:30, T4 19:30-20:00, T5 20:00-20:30), 5 classrooms, 16 professors and 4 specialties. There was one classroom per specialty, while a fifth (alternative) classroom was used to schedule those students that could not be assigned.

We considered that every student would have between one and three examiners, while every professor could assess a minimum of one and a maximum of eight students.

For the experimentation we used a computer with an Intel Celeron processor of 2.16-GHz N2840 CPU and 4 GB of RAM, together with the Windows 10 Home operating system. The data matrices were programmed in Excel, which was linked to LINGO 17 software where we captured the mathematical models and solved them using the exact branch and bound technique. Table 3 shows the conditions and the runtimes used for stage in each experiment. It is worth mentioning that a solution was found in every case for each model.

Table 3 Conditions and runtimes per stage used in the different experiments.

Experiments	Students	Time slots	Classrooms	Professors	Time in seconds to find solution			Total time in seconds
					Model 1 Stage 1	Model 2 Stage 2	Model 3 Stage 3	
1	10	5	5	16	<1	<1	3	<5
2	15	5	5	16	<1	<1	3	<5
3	20	5	5	16	<1	<1	3	<5
4	25	5	5	16	<1	<1	3	<5
5	30	5	5	16	1	1	3	5
6	35	5	5	16	1	1	3	5
7	40	5	5	16	1	1	3	5
8	45	5	5	16	1	1	3	5
9	50	5	5	16	1	2	4	7

Figs. (3) and (4) give a graphic presentation of the level of fulfillment of the conditions established in stage I.

Fig.. (3) shows the behavior of the results in each test as regards the percentage of assigned students on each assessment day. The result of experiments 1–4 indicates that the assessments should be scheduled on the first day (Monday). As of experiment 5, it was necessary to consider the second day (Tuesday) in order to schedule all the assessments. From the results we deduce that, under current conditions, the assignment shall seek to saturate the available time slots on the first assessment day then, as the number of students increases, the assignment will keep on filling in the time slots on the following day.

Moreover, Fig. (4) records the percentage of students assigned at each of the times available for the assessment. The results indicate a clear trend of prioritizing the assignment of the first time slots of each day (Time slot 1 and Time slot 2).

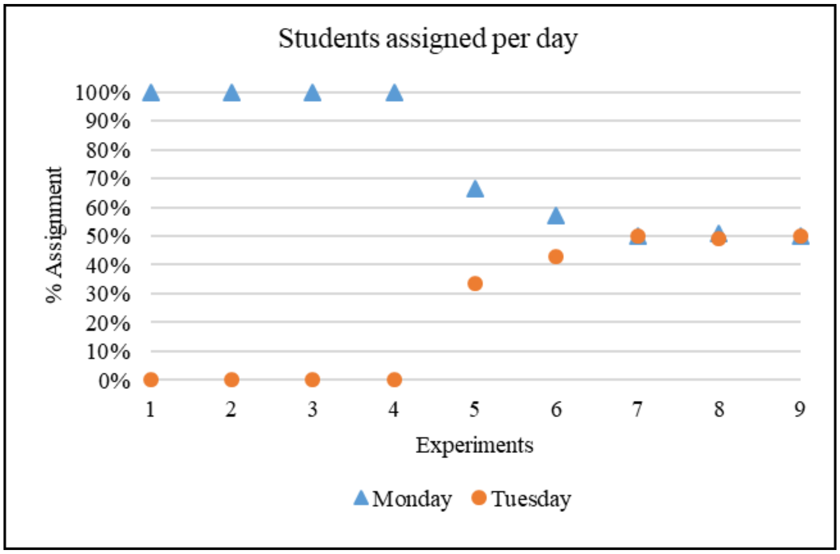

Fig. 3. Graph of students assigned per day.

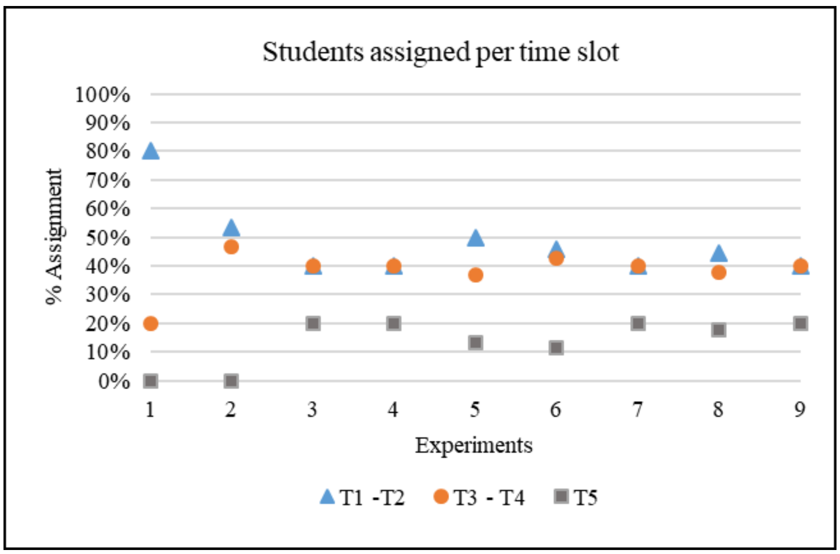

Fig. 4. Graph of students assigned per time slot.

Fig. (5) shows a graph for the fulfillment of the conditions of stage II, corresponding to assigning the highest percentage of students to the classroom of their specialty. The same figure shows that it was not possible in experiments 4, 8 and 9 to assign all the students to their compatible classroom, owing to the fact that the available spaces had already been taken, so the alternative classroom had to be used.

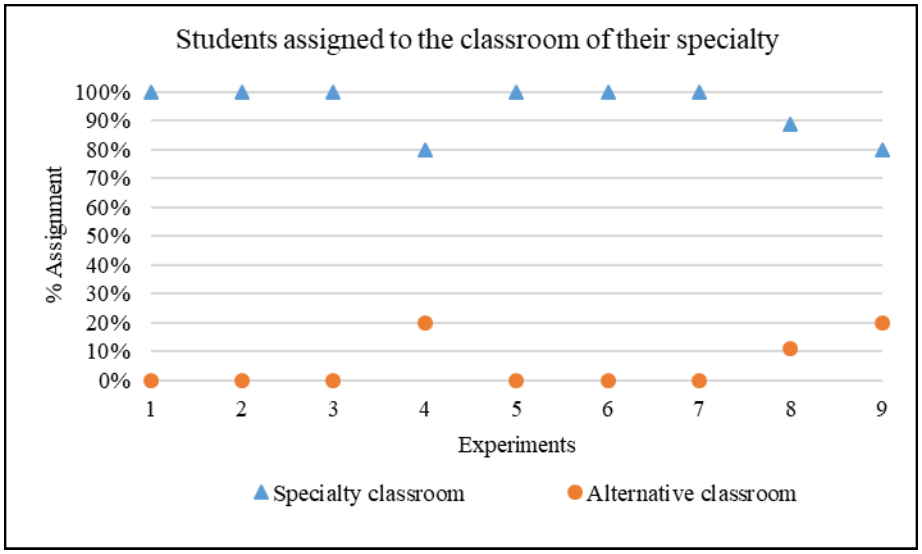

Fig. 5. *Graph of students assigned to the classroom of their specialty.*

Table 4 establishes the percentage of time slots occupied per classroom per day. It must be pointed out that the alternative classroom was used in experiment 4 with the aim of doing all the assessments in a single day while, in experiments 8 and 9 the alternative classroom was required for the purpose of performing the assessments within a maximum of two days. Therefore it can be concluded, that the alternative classroom is used with the intention of assessment of students in the least number of days.

Table 4. Percentage of occupancy of classrooms per day in each experiment.

Experiment	Classroom 1 Specialty 1		Classroom 2 Specialty 2		Classroom 3 Specialty 3		Classroom 4 Specialty 4		Classroom 5 Alternative	
	Monday	Tuesday	Monday	Tuesday	Monday	Tuesday	Monday	Tuesday	Monday	Tuesday
1	**60%**	0%	**60%**	0%	**40%**	0%	**40%**	0%	0%	0%
2	**80%**	0%	**80%**	0%	**80%**	0%	**60%**	0%	0%	0%
3	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	0%	0%
4	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
5	**100%**	**60%**	**100%**	**60%**	**100%**	**60%**	**100%**	**20%**	0%	0%
6	**100%**	**80%**	**100%**	**80%**	**100%**	**80%**	**100%**	**60%**	0%	0%
7	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	0%	0%
8	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**60%**	**40%**
9	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

Fig. (6), (7) and (8) give a graphic presentation of the level of fulfillment of the conditions posed in stage III.
Fig. (6) shows that most of the professors assigned to the assessment have high compatibility with the topic of the student being assessed.
Moreover, Fig. (7) shows that it was possible to assign the maximum number of examiners per student in most of the experiments, except for experiments 4 and 8 where the model assigns more professors with medium compatibility.
Furthermore, in experiments 8 and 9 the limit was reached of eight students to be assessed per professor (see Fig. 8). The number of students and the conditions imposed by the model make it impossible to keep three professors in every assessment.

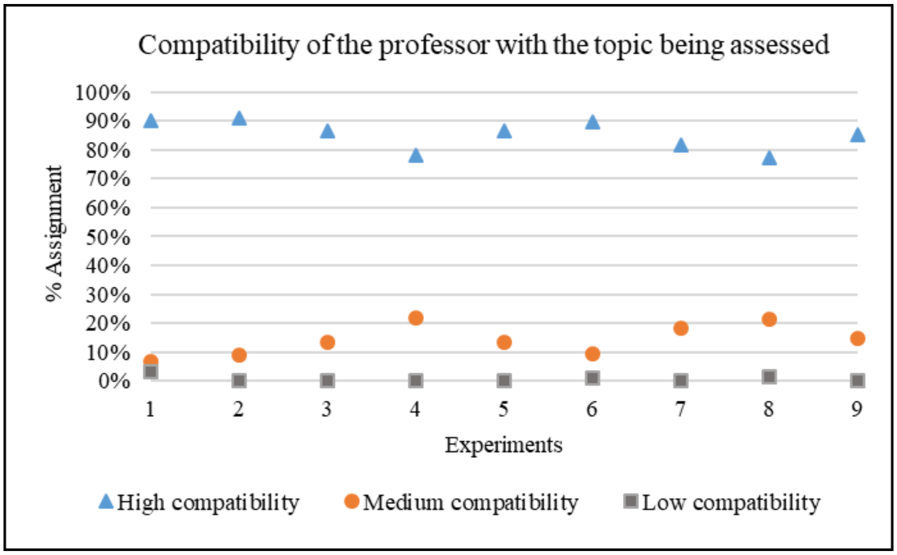

Fig. 6. Graph showing the compatibility of the professor with the topic being assessed.

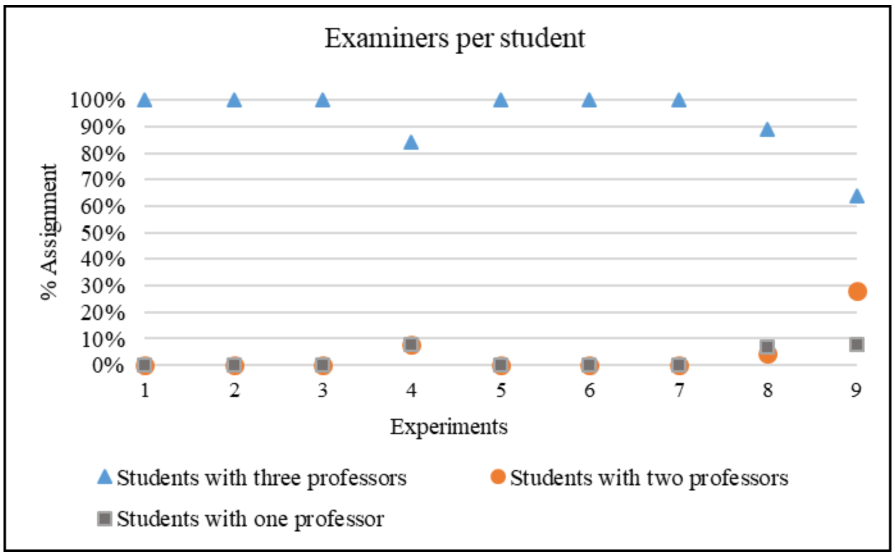

Fig. 7. Graph for examiners per student.

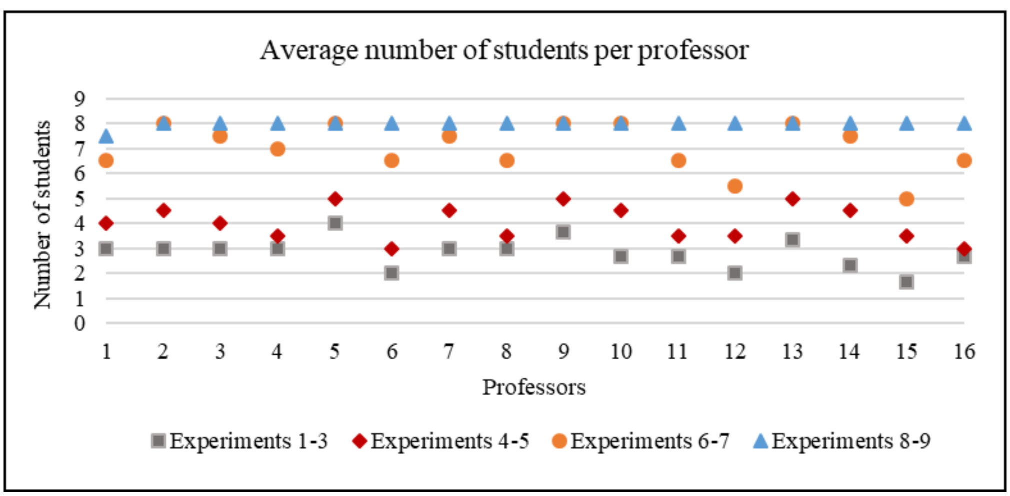

Fig. 8. Graph for the average number of students per professor.

Another important aspect is that the time used in the construction of the timetable is now invested in the analysis of scenarios, thus making it easier to identify the factors that affect the resulting timetable and facilitating decision-making on the part of the person responsible for the examination timetabling.

4.1. Computational results

As we intended to assess the binary mathematical models proposed in different instances, we solved five large-scale problems, considering a range of 50 to 800 students, 5 to 80 classrooms and 16 to 256 professors. The number of binary variables, constraints, Non-Zeros and the times employed in each stage during the search for an optimum solution are given in Table 5.

Table 5. Sizes of problems solved by using mathematical modelling.

		Problems				
		1	**2**	**3**	**4**	**5**
Size of the problem	**Students**	50	100	200	400	800
	Time slots	5	5	5	5	5
	Classrooms	5	10	20	40	80
	Professors	16	32	64	128	256
Binary variables	**Stage I**	250	500	1,000	2,000	4,000
	Stage II	250	1,000	4,000	16,000	64,000
	Stage III	800	3,200	12,800	51,200	204,800
	Total	**1,300**	**4,700**	**17,800**	**69,200**	**272,800**
Constraints	**Stage I**	56	106	206	406	806
	Stage II	76	151	301	601	1,201
	Stage III	533	1,065	1,489	2,977	No solution found
	Total	**665**	**1,322**	**1,996**	**3,984**	---------
Non zeros	**Stage I**	750	1,500	3,000	6,000	12,000
	Stage II	750	3,000	12,000	48,000	192,000
	Stage III	5,600	22,398	89,592	358,360	No solution found
	Total	**7,100**	**26,898**	**104,592**	**412,360**	----------
Time in seconds to find the solution	**Stage I**	1	1	2	4	15
	Stage II	2	2	3	10	21
	Stage III	4	5	11	42	No solution found
	Total	**7**	**8**	**16**	**56**	-----------

In stage III (assignment of professors) of problem number 5, an optimum solution could not be found after a run of 4 hours, so the decision was made to stop the search.

One of the merits of the strategy for solving the problem of the construction of examination timetables by stages is the huge decrease in the number of binary variables and, in consequence, of the feasible solution space. Table 6 gives a comparison of the number of variables required to solve the problems, considering only one mathematical model or the three models for the step-by-step strategy. The last part of the table gives the percentage for the reduction of variables that, as a result of the new strategy being implemented, bounds the feasible solution space of each problem.

Table 6. Comparison of binary variables between 1 and 3 mathematical models.

Mathematical models	***Problems***				
	1	**2**	**3**	**4**	**5**
1 Model	20,000	160,000	1,280,000	10,240,000	81,920,000
3 Models (Three stages)	1,300	4,700	17,800	69,200	272,800
% Reduction of variables	**93.50%**	**97.06%**	**98.61%**	**99.32%**	**99.67%**

5. Conclusions

This article presented a new strategy for the definition of examination timetables over three stages of assignment and using binary variables with two indexes. With this strategy, the problem is decomposed into more than one mathematical model, thus getting solutions in a shorter time, instead of solving the entire problem in a single step.

To speed up the search for solutions, our strategy was to decompose the original problem into three mathematical models, which considerably cut down on the number of possible combinations; we also used subsets that group the students according to specialty and time slot, while the coefficients of the objective functions were determined for each model.

The analysis, model construction and solution process is exemplified by using examination timetabling as a specific application in the Tecnológico Nacional de México in Celaya. This administrative problem used to be solved by hand using trial and error, which took up a lot of the postgraduate coordinator's time; whereas with the proposal developed in this study, the process can be automated and they can now invest their time in analyzing scenarios.

The proposal described in this article can be used by other educational institutions, particularly ones that form part of the system of Technological Institutes of Mexico, for scheduling their examination periods. One interesting area for future research would be to study the impact of this strategy on the analysis of other combinatory problems.

Acknowledgements

We would like to thank the Consejo Nacional de Ciencia y Tecnología (Mexican Council of Science and Technology - CONACYT) for financing this research, under the Unique Resume registration number: 375569.

References

[1] Even, S., Itai A., Shamir A.: On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing, 5(4), 691–703 (1976).
[2] Bardadym V. A.: Computer-aided school and university timetabling: the new wave. In: Burke E, Ross P editors. Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science, 1153, 22-45 (1996).
[3] Caramia, M., Olmo, P. D., Italiano, G. F., Informatica, D., & Produzione, S.: New Algorithms for Examination Timetabling. International Workshop on Algorithm Engineering, 230–241 (2000).
[4] Casey, S., & Thompson, J.: GRASPing the Examination Scheduling Problem. Practice and Theory of Automated Timetabling IV, 2740, 232–244 (2003).
[5] Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M.: Investigating Ahuja–Orlin's large neighbourhood search approach for examination timetabling. OR Spectrum, 29(2), 351–372 (2007a).
[6] Di Gaspero, L., & Schaerf, A.: Tabu Search Techniques for Examination Timetabling. International Conference on the Practice and Theory of Automated Timetabling III, 104–117 (2001).
[7] Abdullah, S., Ahmadi, S., Burke, E. K., Dror, M., & Mccollum, B.: A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. Journal of the Operational Research Society, 58(11), 1494–1502 (2007b).
[8] Pais, T. C., & Amaral, P.: Managing the tabu list length using a fuzzy inference system: An application to examination timetabling. Annals of Operations Research, 194(1), 341–363 (2012).
[9] June, T. L., Obit, J. H., Leau, Y., & Bolongkikit, J.: Implementation of Constraint Programming and Simulated Annealing for Examination Timetabling Problem. Computational Science and Technology, 481, 175–184 (2019).
[10] Cheong, C. Y., Tan, K. C., & Veeravalli, B.: A multi-objective evolutionary algorithm for examination timetabling. Journal of Scheduling, 12(2), 121–146 (2009).
[11] Pillay, N., & Banzhaf, W.: An informed genetic algorithm for the examination timetabling problem. Applied Soft Computing Journal, 10(2), 457–467 (2010).
[12] Leite, N., Fernandes, C. M., Melício, F., & Rosa, A. C.: A cellular memetic algorithm for the examination timetabling problem. Computers and Operations Research, 94, 118–138 (2018).
[13] Burke, E., Bykov, Y., & Petrovic, S.: A Multicriteria Approach to Examination Timetabling. International Conference on the Practice and Theory of Automated Timetabling, 2079, 118–131 (2001).

[14] Mumford, C. L.: A multiobjective framework for heavily constrained examination timetabling problems. Annals of Operations Research, 180(1), 3–31 (2010).
[15] Lei, Y., Shi, J., & Yan, Z.: A memetic algorithm based on MOEA/D for the examination timetabling problem. Soft Comput, 22(5), 1511–1523 (2018).
[16] Kendall, G., & Hussin, N.: An investigation of a tabu-search-based hyper-heuristic for examination timetabling. Multidisciplinary Scheduling: Theory and Applications, 309–328 (2005).
[17] Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R.: A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research, 176(1), 177–192 (2007).
[18] Sabar, N. R., Ayob, M., Qu, R., & Kendall, G.: A graph coloring constructive hyper-heuristic for examination timetabling problems. Applied Intelligence, 37(1), 1–11 (2012).
[19] Burke, E. K., Kendall, G., Misir, M., & Özcan, E.: Monte Carlo hyper-heuristics for examination timetabling. Annals of Operations Research, 196(1), 73–90 (2012).
[20] Demeester, P., Bilgin, B., De Causmaecker, P., & Berghe, G. V.: A hyperheuristic approach to examination timetabling problems: Benchmarks and a new problem from practice. Journal of Scheduling, 15(1), 83–103 (2012).
[21] Qu, R., Pham, N., Bai, R., & Kendall, G.: Hybridising Heuristics within an Estimation Distribution Algorithm for Examination Timetabling. Applied Intelligence, 42(4), 679–693 (2015).
[22] Gogos, C., Alefragis, P., & Housos, E.: A Multi-Staged Algorithmic Process for the Solution of the Examination Timetabling Problem. Annals of Operations Research, 194(1), 203–221 (2012).
[23] Abdul-Rahman, S., Burke, E. K., Bargiela, A., McCollum, B., & Özcan, E.: A constructive approach to examination timetabling based on adaptive decomposition and ordering. Annals of Operations Research, 218(1), 3–21 (2014).
[24] Mohmad Kahar M. N., & Kendall, G.: A great deluge algorithm for a real-world examination timetabling problem. Journal of the Operational Research Society, 66(1), 116–133 (2015).
[25] Woumans, G., De Boeck, L., Beliën, J., & Creemers, S.: A column generation approach for solving the examination-timetabling problem. European Journal of Operational Research, 253 (2016).
[26] MirHassani, S. A.: Improving paper spread in examination timetables using integer programming. Applied Mathematics and Computation, 179(2), 702–706 (2006).
[27] Al-Yakoob, S. M., Sherali, H. D., & Al-Jazzaf, M.: A mixed-integer mathematical modeling approach to exam timetabling. Computational Management Science, 7(1), 19–46 (2010).
[28] McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., & Qu, R.: A new model for automated examination timetabling. Annals of Operations Research, 194(1), 291–315 (2012).
[29] Cataldo, A., Ferrer, J. C., Miranda, J., Rey, P. A., & Sauré, A.: An integer programming approach to curriculum-based examination timetabling. Annals of Operations Research, 258(2), 369–393 (2017).
[30] Kasm, O. A., Mohandes, B., Diabat, A., & El Khatib, S.: Exam Timetabling with Allowable Conflicts within a Time Window. Computers and Industrial Engineering, 127, 263–273 (2019).
[31] Qu, R., Burke, E. K., Mccollum, B., Merlot, L. T. G., & Lee, S. Y.: A survey of search methodologies and automated approaches for examination timetabling. Computer Science Technical Report No. NOTTCS-TR-2006-4 UK (2006).