Automated Activity Scheduling Tools for Improving Learning and Evaluation of Cybersecurity Competencies in Computer Engineering Courses

https://doi.org/10.3991/ijet.v18i08.34879

Borja Bordel(✉), Ramón Alcarria, Tomás Robles
Universidad Politécnica de Madrid, Madrid, Spain
borja.bordel@upm.es

Abstract—In the last ten years, many different innovative learning methodologies have been proposed for engineering education. In those new approaches, students learning, and evaluation are usually supported by a catalogue of creative activities, practices, and presentations which may be very numerous if competencies to be acquired are heterogeneous and must cover a large knowledge area, such as in cybersecurity courses. In addition, when a high number of students are enrolled in the course, deadlines must be strict, to enable professors to evaluate all students and activities properly and on time. Due to this requiring schedule, students often cannot work all competencies with the expected depth, and their learning decreases. New instruments are needed to facilitate the learning of the students and the evaluation process are needed. Therefore, in this paper we proposed a new automated scheduling tool, based on graph theory, to fill this gap. The tool employs graph coloring algorithms to calculate all possible schedules for evaluation activities. Students may freely choose the schedule that best fits their personal situation, learning progress, background, etc. using a web portal where all solutions from the coloring algorithm are displayed. This tool was deployed in the Computer Engineering degree at Universidad Politécnica de Madrid, Spain. A pilot experience was conducted for three years (2020–2022) in the context of a cybersecurity course to analyze how this new tool improves the learning and the evaluation process of students. Results show both academic results and the students' and professor' satisfaction improve in a significant manner.

Keywords—cybersecurity, computer engineering, competencies, learning by doing, evaluation, automated scheduling

1 Introduction

In the last ten years, many different innovative learning and teaching methodologies have been proposed for engineering education [1]. In general, these new methodologies focus on generating a high added value for students in synchronous in-person sessions and activities. For example, in new proposals such as Flipped Classroom [2] or Research-based learning [3], in-person sessions consist of presentations, discussion,

knowledge creation actions, and evaluation activities. In the context of engineering education, the most relevant methodologies are actually based on the 'Learning by Doing' paradigm [4], such as Project-based learning [5], challenge-based learning [6] or service learning [7]. In this paradigm, student learning and evaluation, are supported by a catalogue of creative activities, practices, and presentations aimed to work specific abilities and competencies through the definition and creation of tangible engineering products.

In "Learning by Doing" methodologies, the number of products and activities that students must perform may be increased in order to cover all the planned competencies with the expected depth level [8]. Even, the number of activities can increase if competencies to be acquired are heterogeneous and must cover a large knowledge area, such as in cybersecurity courses. Thus, it is common for cybersecurity students to face hard schedules with tens of activities and presentations in one single term. Also, when a high number of students are enrolled in cybersecurity courses (as usually happens in computer engineering degrees), deadlines cannot be flexible. Although flexibility may help students improve their learning, in courses with hard schedules with tens of planned activities and more than one hundred enrolled students, deadlines must be strict, to allow professors to present, discuss and evaluate all students and activities properly and on time [9].

Because of this demanding schedule, cybersecurity students often cannot work all competencies in the expected depth [10]. Personal appointments, work obligations, background deficits, etc., may prevent students to invest the needed time to go deeper in the proposed activities, and finally their learning decreases. Personalized learning strategies could mitigate this situation, but it is almost impossible for professors to manually customize the schedule for hundreds of students [11] without producing conflicts among students (because they may feel that some students are receiving a better schedule), problems with competencies (because learning activities may follow a specific order, or some activities may be incompatible), or agenda problems (all activities must be scheduled within the planned sessions and slots).

In conclusion, new instruments are needed to facilitate the students in working on evaluation activities, improve the learning personalization, and help professors organize and schedule all the planned activities. Therefore, in this paper we proposed a new automated scheduling tool to help students and professors deal with evaluation and learning activities. The tool is based on graph theory. It represents activities as vertices and restrictions (activities that are incompatible and cannot have the same deadline or take place in the same slot) as edges. The tool employs graph coloring algorithms to calculate all possible schedules for evaluation activities, given the restrictions. Colors represent available slots for deadlines and activities, according to the professors' agenda. Students may choose the schedule that best fits their personal situation, learning progress, background, etc. using a web portal where all solutions from the coloring algorithm are displayed. Once any student book a particular schedule for his evaluation activities, this schedule is removed from the platform. Future improvements may allow several students to have the same schedule, but the currently employed software does not have that feature: all schedules must be different.

A pilot experience was conducted for three years (2020–2022) in order to analyze how this new tool improves student learning and the global evaluation process. The tool was deployed at Universidad Politécnica de Madrid, Spain, in the context of a cybersecurity course belonging to the third course of the Computer Engineering degree.

The remainder of the paper is organized as follows. Section 2 analyzes previous experiences with automated activity scheduling in higher education. Section 3 describes from a technical perspective the proposed new tool. Section 4 introduces the methodology for the experimental pilot experience. Section 5 presents and discusses the results of the pilot experience, and Section 6 concludes the paper.

2 State of the art

Automated activity scheduling tools in higher education have been studied from different perspectives. Although some very specific use cases may be found, for example, applications to calculate the best moment (week) for scheduling an engineering syllabus course [12], in general, reported solutions are focused on two general problems: class scheduling [13] and, mainly, exam scheduling [14].

Regarding class scheduling mechanisms, proposed algorithms and solutions usually consider two dimensions: room allocation and time slot allocation. Some authors have explored ad-hoc allocation and scheduling algorithms [15] with embedded restrictions taken from their institutions and courses. In the last five years, genetic algorithms [37][38] have become the popular technology in this area. This approach is very effective, but it is not flexible and cannot be easily adapted to other scenarios. On the other hand, optimization models based on visual tools and/or linear programming have been reported [25]. However, although this approach has the potential to be applied to other institutions, models are still specifically designed for certain courses. Some simple algorithms, implemented employing standard software such as Excel, have been used to schedule timetables in general scenarios [16], but the restrictions that can be applied are very simple and the performance of the tool is limited.

The main open challenge in automated class scheduling is the integration of "soft restrictions" into the optimization algorithm. For example, conditions related to the number of students in each room [17] or non-usual break periods [18]. To address this problem, many different scheduling and optimization techniques have been applied to class scheduling: from Particle Swarm Optimization (PSO) [19] and integer programming [20], to hyperheuristics [21], fix-and-optimize metaheuristic [22] and several different algorithms such as the Great deluge algorithm [22] or genetic algorithms [23]. However, all these techniques have poor performance and false optimum solutions are usually calculated.

In a totally different approach, some authors have studied totally random class scheduling as a potential solution [24]. But the results are not solid enough to conclude if this approach is successful or not.

The second problem where automated scheduling tools are applied in the context of higher education is exam scheduling [14]. In the state of the art, six different techniques

for exam scheduling have been reported; some of them are similar to the ones employed for class scheduling.

Genetic algorithms [26] are one of the most widely used technologies. This approach enables professors to choose between different examen schedules that are solutions to the optimization problem [27], but algorithms must be adapted for specific institutions and cannot be easily extended to other scenarios. The same technology has also been used to coordinate exam vigilance responsibilities between professors participating in the same course [28].

Metaheuristics are also employed in the scheduling of exams, such as the Tabu algorithm [29]. It is a very old approach, with poor performance in quite complex scenarios because of the cumbersome way in which problems are described. A similar problem can be observed in tools based on the quadratic assignment problem [30], where solutions are arranged in five different phases until the final optimum schedule is generated.

Moreover, probabilistic techniques such as simulated annealing [31] have been reported as well. Although the solutions in the simulated annealing tools are not fully compliant with the initial restrictions and the scenario description.

As in class scheduling, linear programming has also been used for exam scheduling [32]. This approach is very effective, but solutions are parametric, and new algorithms to assign optimum values to parameters are needed. Then, domain-specific algorithms are used, and final tools cannot be easily implemented in other use cases.

The last technique employed for exam scheduling in the state of the art is graph theory. Many different approaches within graph theory can be employed to find optimum schedules, but graph coloring is the most efficient and popular [33]. The main problem is the high computational cost of this approach, but since no real-time requirements must be met, this inconvenience is not critical. In this paper, we are using this approach and technology to develop our scheduling tool.

Finally, hybrid tools, where more than one technology is implemented, can be found. These schemes are not common as the final performance does not improve the results of the individual techniques in a significant way. But some solutions with several refinement cycles, each based on a different technique, have been described [34].

In general, existing class and exam scheduling tools can generate only one schedule, as a solution to a given problem. However, in our tool the approach is different. The proposed algorithm computes all possible solutions and students are free to individually choose the schedule that better fits their personal situation.

3 Automated activity scheduling tool

In order to improve the students learning in cybersecurity competencies and the entire evaluation process (so students have enough time to show their skills and professors are able to analyze their knowledge and abilities carefully), a new tool for the automated (and personalized) scheduling of activities is proposed. Section 3.1 describes the underlying mathematical problem based on the graph theory to be solved, and how the catalogue of possible schedules is calculated using a coloring algorithm. Besides, Section 3.2 presents the online software tool employed by students to choose their

personalized schedule, as well as by professors to configure the available slots and potential deadlines for activities.

3.1 Mathematical problem and schedule calculation: graph theory

In a cybersecurity course, a set A of N different evaluation activities a_i must be scheduled (1).

$$A = \{a_i \quad i = 1, \ldots, N\} \tag{1}$$

These activities consist of two products to be submitted: a written report and a synchronous public presentation. Typically, deadlines and time slots for these two products are not independent, and reports must be submitted to the institutional Learning Management System (LMS) at least twenty-four (24) hours before the appointment for public presentation. As a result, for each activity a_i the automated scheduling tool only must manage the time slots for public presentations, as deadlines for reports may be directly obtained from those appointments.

In general, professors may employ K different time slots γ_k in evaluation activities (public presentations), distributed along the entire semester (2). To generate a new potential activity schedule, a time slot c_i must be associated to every activity a_i but according to some given restrictions e_j (3). Function $f_{sch}(\cdot,\cdot)$ makes that association.

$$\Gamma = \{\gamma_k \quad k = 1, \ldots, K\} \tag{2}$$

$$f_{sch}(a_i, \{e_j\}) = c_i \quad c_i \in \Gamma \tag{3}$$

In this context, restrictions e_j and activities a_i define a graph G (4). In this graph, the vertices $V(G)$ are the activities a_i to be evaluated in the cybersecurity course (5); and the edges $E(G)$ are restrictions e_j between activities a_i (6), i.e., incompatibility relations between activities that cannot be evaluated in the same time slot. In this context, two activities a_i and a_r are connected (are adjacent, using the graph theory terminology) through an edge $e_j = (a_i, a_r)$ if those activities cannot be associated to the same time slot (7).

$$G = (V, E) \tag{4}$$

$$V(G) = A = \{a_i \quad i = 1, \ldots, N\} \tag{5}$$

$$E(G) = \varepsilon = \{e_j \quad j = 1, \ldots, M\} \tag{6}$$

$$c_i \neq c_r \Rightarrow \exists\, e_j \in E(G) \;\vdots\; e_j = (a_i, a_r) \tag{7}$$

The resulting graph has no loops, as they have no physical sense in our scenario. In addition, the graph is undirected since the restrictions are totally symmetric and bidirectional. Moreover, there are no multiple edges as they would be redundant and do

not add any additional information. Then, the graph modeling our problem is simple. Furthermore, since the number of vertices and edges is finite (evaluation activities cannot be infinite), the graph is finite (and simple). Figure 1 shows a small example of a graph representing the scheduling problem.

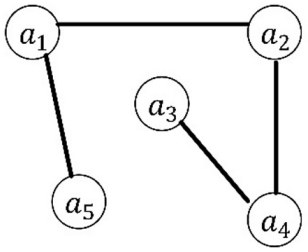

Fig. 1. Small graph representing an example of a scheduling problem

In order to make the analysis and processing of the resulting graph easier, in our software scheduling tool, it is represented using the adjacency matrix $\mathcal{M}$ (8). In this binary matrix, element $m_{i,r}$ is equal to the unit if vertices a_i and a_r are adjacent in graph G. On the contrary, $m_{i,r}$ is zero.

$$\mathcal{M} = \begin{pmatrix} 0 & m_{1,2} & \cdots & m_{1,N} \\ m_{2,1} & 0 & \cdots & m_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ m_{N,1} & m_{N,2} & \cdots & 0 \end{pmatrix} \quad (8)$$
$$being\ m_{i,j} = m_{j,i} \quad \forall i, j \in [1, N]$$

Each activity a_i in the graph is labeled with three pieces of information. Namely:

- First, the complexity level μ_i of the activity. This refers to the time required to finalize the activity according to the given instructions and/or the expected elaboration level of the submitted products.
- Second, the list ℓ_i of cybersecurity competencies l_i^s to be worked with the activity. To simplify the mathematical analysis of this list, competencies are represented by natural numbers.
- Finally, the intensity or depth level d_i^s with which each competency in the list ℓ_i must be worked. All these levels are also grouped in a list D_i. According to ACM (Association for Computing Machinery) recommendations for Computer Engineering degrees [35], this level is represented using numbers in the interval [1,3].

These labels are assigned through an application λ (9), which in the proposed tool is manually generated by professors using the online software platform (see Section 3.2).

$$\begin{aligned}\lambda(a_i) &= \{\mu_i, \ell_i, D_i\} \\ D_i &= \{d_i^s \quad s = 1, \ldots, S_i\} \\ \ell_i &= \{l_i^s \quad s = 1, \ldots, S_i\}\end{aligned} \tag{9}$$

Using these labels, it is possible to automatically determine if two activities a_i and a_r must be adjacent or not. And then, it is possible to obtain the adjacent matrix. In this work, we are considering two cybersecurity activities cannot have the same deadline if (i) both activities have an elevated complexity level; (ii) they work competencies too different; or (iii) they work several competencies with a high intensity.

These conditions are represented using the Kronecker's delta function $\delta[n]$ (10), the discrete Heaviside step function $u[n]$ (11) and three positive real thresholds for the maximum average complexity μ_{th}, the maximum average intensity d_{th} and the maximum average difference between competences L_{th}. Those parameters can be controlled by professors through the online tool (see Section 3.2), which can later automatically obtain the adjutancy matrix (12).

$$\delta[n] = \begin{cases} 1 \quad if \quad n = 0 \\ 0 \; otherwise \end{cases} \tag{10}$$

$$u[n] = \begin{cases} 1 \; if \quad n < 0 \\ 0 \; otherwise \end{cases} \tag{11}$$

$$\begin{aligned} m_{i,j} = m_{j,i} = \delta \Bigg[& u\left[\frac{1}{2}(\mu_i + \mu_j) + \mu_{th}\right] + u\left[\frac{1}{S_i}\sum_{s=1}^{S_i} d_i^s + \frac{1}{S_j}\sum_{s=1}^{S_j} d_j^s - d_{th}\right] \\ & + u\left[\frac{1}{S_i S_j}\sum_{s=1}^{S_i}\sum_{r=1}^{S_j}\left|l_i^s - l_j^r\right| - L_{th}\right]\Bigg] \end{aligned} \tag{12}$$

Once the entire graph is constructed, considering it is simple and finite, possible schedules can be obtained through a graph coloring process with K colors from the color universe Γ.

Typically, the graph coloring problem is looking for a function $g_{color}(\cdot)$, so every vertex (activity a_i) is associated with a color (time slot) c_i, but adjacent activities are labeled always with different colors (13). Typically, in addition, coloring algorithms look for the optimum solution, where the minimum number of colors is employed to label all vertices (activities). Actually, the minimum number of colors required to color a graph is a property known as the chromatic number, $\chi(G)$. In our case, employing the minimum number of time slots in evaluation actions is also desired. But coloring algorithms should produce all possible solutions, so students may choose the one that best fits their personal situation.

$$g_{color} : V(G) \to \Gamma \quad being \quad g_{color}(a_i) \neq g_{color}(a_r) \;\; if\,(a_i, a_j) \in \mathcal{E} \tag{13}$$

On the other hand, an additional restriction is also added. Each time slot (color) can only be associated with a maximum of T activities (vertices). This condition will be added to coloring algorithms and also to the online platform so that conflicting schedules are removed from the platform as soon as the maximum number of activities in any time slot is reached.

Two different coloring algorithms, producing different solutions (schedules), are employed to generate all possible activity schedules and feed the online platform.

By default, a Greedy (or sequential) algorithm is employed. Algorithm 1 describes the implementation selected for this computational solution.

Algorithm 1: Greedy coloring algorithm

Input: List of activities $\sigma = \{\sigma_i \;\; \forall i \in [1, N]\}$
List of colors $\Gamma = \{\lambda_r \;\; \forall r \in [1, K]\}$
Output: Coloring function $g_{color}(\cdot)$

$g_{color}(\sigma_1) = \lambda_1$
Define $\Phi = \{\phi_m \;\; \forall m \in [1, K]\}$ a natural array
$\phi_1 \leftarrow 1$

for each $i \in [2, N]$ **do**
  for each $j \in [1, i-1]$ **do**
    if $(\sigma_i, \sigma_j) \notin E(G)$ **and** $\phi_1 < T$ **do**
      $g_{color}(\sigma_i) = \lambda_j$
      Increment ϕ_j in one unit
      break
    end if
  end for
end for

Algorithm 2: Brute-force search algorithm

Input: List of activities $\sigma = \{\sigma_i \;\; \forall i \in [1, N]\}$
List of colors $\Gamma = \{\lambda_r \;\; \forall r \in [1, K]\}$
Output: All valid coloring function $\{g^m_{color}(\cdot)\}$

Define integer r
$r \leftarrow 1$

function color [input: activity σ_m] **do**
  for each $i \in [1, K]$ **do**
    $g^r_{color}(\sigma_m) = \lambda_i$
    if $\sigma_m \neq \sigma_N$ **do**
      color [s_{m+1}]
    end if
    Increment r in one unit
  end for
end function

Basically, vertices (activities) must be ordered in a list, as well as colors. Then, the first vertex is labeled with the first color. The second vertex is labeled with the first color too, unless it is adjacent to the first activity or more than T activities are already labeled with the first color, so it is labeled with the second color. Then, the third activity is labeled with the first color too, unless it is adjacent to the first vertex or more than T activities are already labeled with the first color; then it is labeled with the second color, unless it is adjacent to the second activity or more than T activities are already labeled with the second color; so, it is labeled with the third color. The process is repeated for all N activities in the list. This algorithm is independent from the order of colors in the list, but generates a totally different solution depending on how activities are ordered in the initial list. Then, the number of potential schedules Ω_{sq} is factorial with the number of activities N (14). Our tool is able to explore

all possibilities and offer all the resulting schedules to students through the online platform.

$$\Omega_{sq} = N! \tag{14}$$

An exhaustive analysis of all possibilities requires a large processing time. Therefore, before running such analysis, if the number of students to be evaluated allows it, only some special ordered lists of vertices are considered. Specifically, three options are implemented in our automated scheduling tool: random list, Welsh-Powell algorithm (where activities are ordered in a decreasing order, starting from the one with a higher order, i.e., with a higher number of adjacent vertices or higher number or restrictions) and Matula-Marble-Isaccson algorithm (where the list of vertices follows exactly the opposite order than in the Welsh-Powell algorithm). Using these lightweight algorithms, up to five different personalized schedules may be generated.

Anyway, as usually N is not higher than fifteen activities, and our scenario has no real-time scheduling requirements, and exhaustive study is computationally feasible.

Finally, in some rare cases, the Greedy algorithm may not generate enough different schedules for all students. Additionally, the number of available schedules on the online platform should be at least 10% higher than the number of students, so all of them can choose among different options. In those cases, the second coloring algorithm is triggered: brute-force search.

In the brute-force search algorithm, all possible assignment of K colors are evaluated. And then those that do not meet the given restrictions are removed. This approach is very impractical and computationally expensive. Therefore, it is only employed when the Greedy algorithm cannot satisfy the required number of possible schedules. In addition, as soon as the brute-force search produces the required number of schedules, the algorithm is stopped, because of its great computational cost. Algorithm 2 shows the selected implementation for the brute-force search algorithm in the proposed activity scheduling tool.

3.2 Online platform for a personalized schedule

The proposed graph problem and coloring algorithms are defined and triggered via an online platform. In addition, students may use the same platform to choose their personalized schedule among all possible solution generated by the coloring algorithms. Finally, the proposed platform is connected to the institutional LMS, so activities and submissions are automatically configured according to the schedule selected by each participant.

Figure 2 shows the workflow and architecture for the proposed online platform.

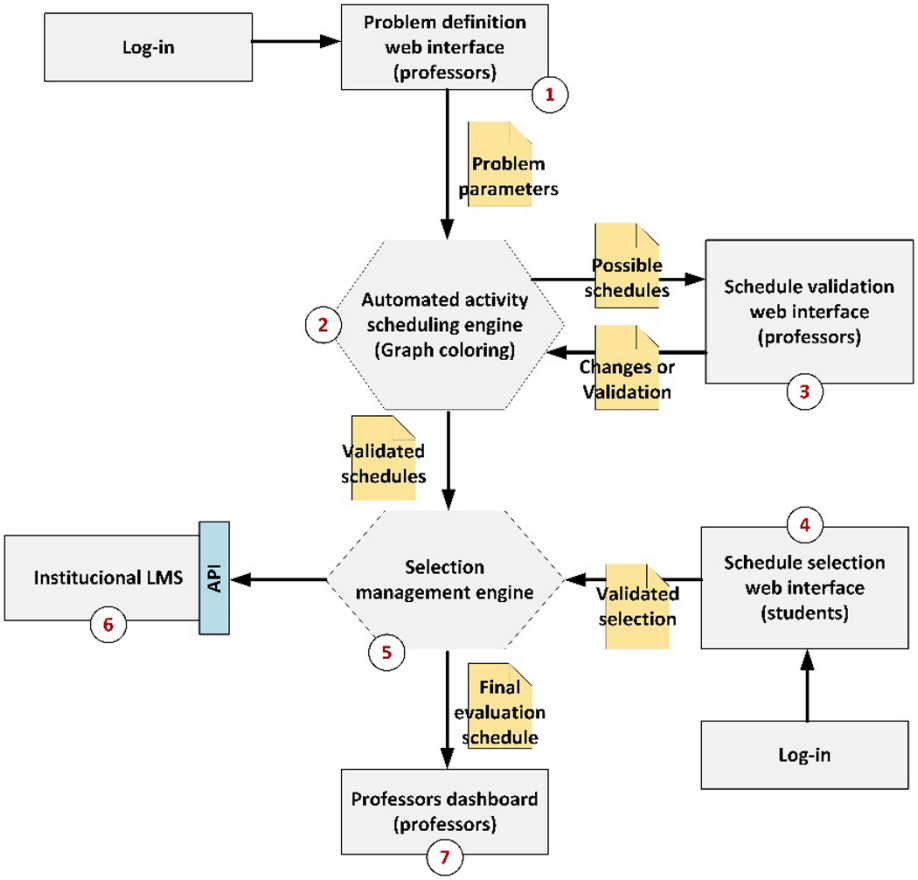

Fig. 2. Workflow and architecture of the proposed scheduling tool

The online platform opens with a log-in page, where professors and/or students are identified. Both profiles could see the subject in a personal space within the online platform. But students may only operate with the platform after the professors have made the required configurations.

The process starts with the professors defining the parameters for the course ①: number of activities (N), difficulty levels (μ_i), competencies associated with each activity (ℓ_i), intensity of each competence (D_i), available slots for evaluation (K) and thresholds for slots in terms of number of activities to be evaluated (T), difficulty level (μ_{th}), similarity in competences (L_{th}) and intensity in competencies (d_{th}). Figure 3 shows the proposed interface for this step.

Then, professors can trigger the automated scheduling process. After the computation is completed in the specific engine ②, professors may review the obtained schedules, remove some of them, ask the platform for a new calculation from scratch, modify the problem parameters, and, eventually, approve schedules ③. Before final approval, professors have to define a deadline for students to choose the evaluation activity schedule.

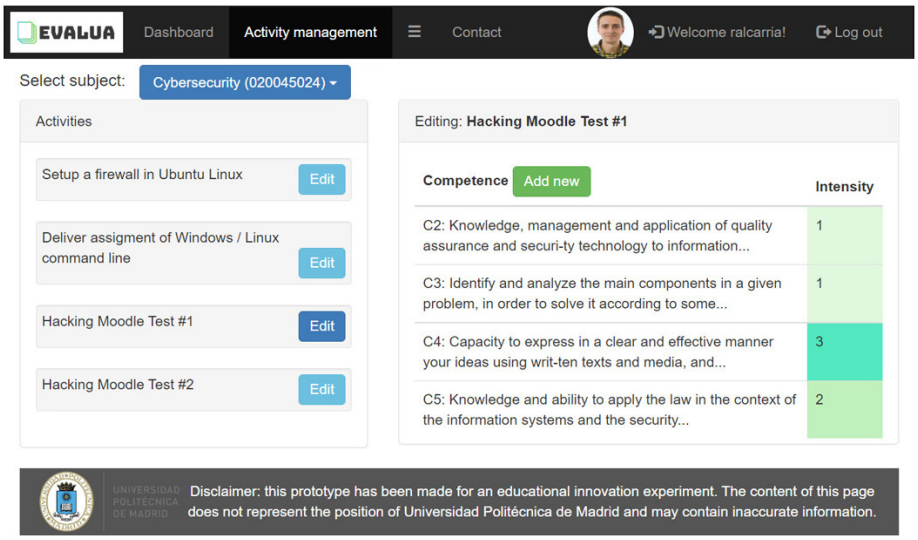

Fig. 3. Problem definition web interface

Once the schedules are approved, students can access to the online space for the cybersecurity course and choose the schedule that best fits their situation ④. Students may validate their selection, and this validation cannot be changed anymore. The system follows a FIFO (First-in First-Out) approach, so students can only choose among the schedules that are already available when accessing the platform.

Figure 4 shows the interface for students in the online tool.

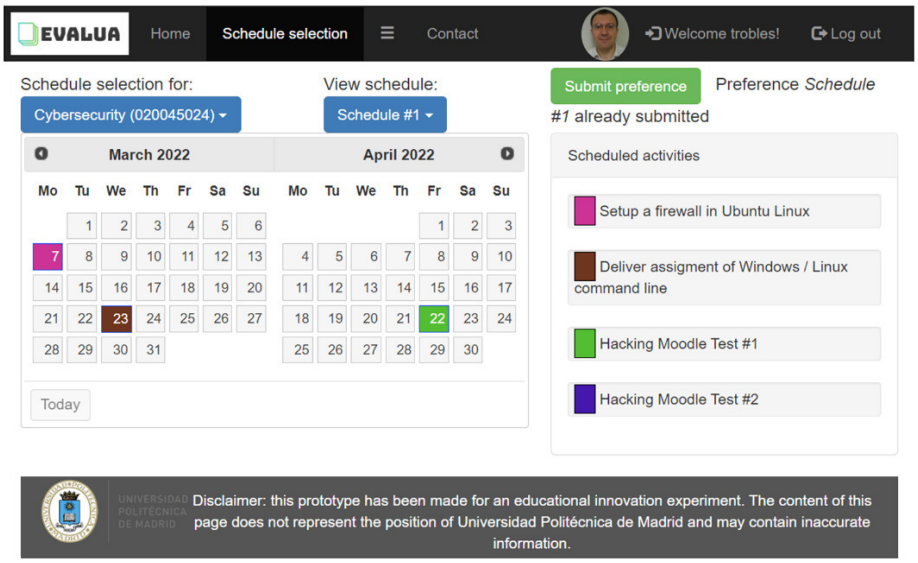

Fig. 4. Web interface for schedule selection

After every student's selection, the chosen schedule is removed from the platform, together with all other incompatible schedules ⑤. A schedule management engine performs those actions. In particular, all other schedule where activity a_i is labeled with color c_i, but T schedules with the same labeling have been already validated, are removed. Additionally, the validation process triggers an automatic configuration in the institutional LMS ⑥. Spaces with the proper deadlines for submissions are automatically configured according to the student's selection in the institutional LMS. Therefore, participants can easily submit their report (and other products) in digital format and always according to their personalized activity schedule. To do that, the public LMS (Moodle) API (Application Programming Interface) was employed.

After the deadline for students to choose their schedule, participants with no validation selection are randomly assigned to an evaluation schedule by the own platform. The student will receive an email with the information. Additionally, professors can get all the information about the scheduled slots, activities to be evaluated, etc. using a dashboard in the same online platform ⑦. Figure 5 shows the professor dashboard.

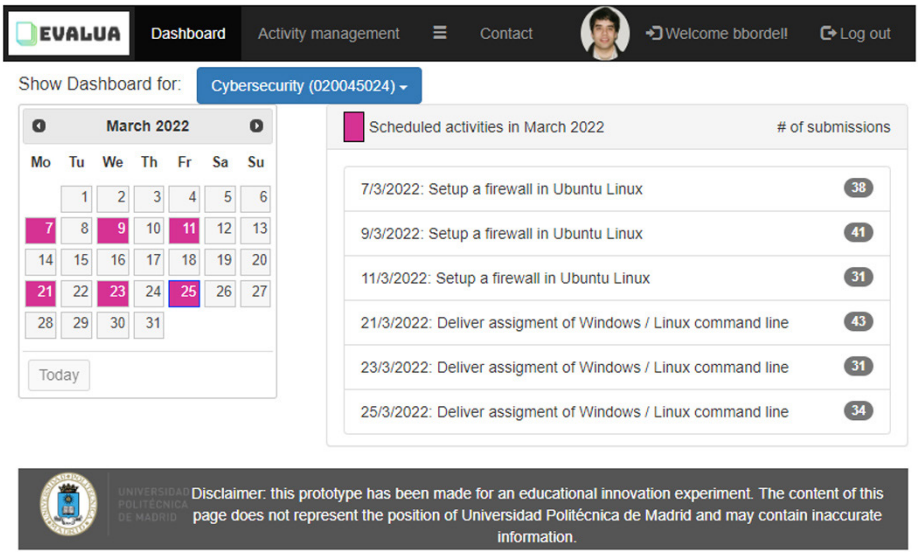

Fig. 5. Professor dashboard

Professors still may manually configure and modify due dates for all evaluation activities using the LMS dashboard. Professors may use this flexibility in extending or revising the assignment due dates during the entire semester, to address unexpected situations.

The entire platform is based on web technologies, specifically JavaScript web programming. This approach allows students to operate with the platform through any kind of connected device and facilitates the interconnection of our new tool with the existing institutional LMS (based on Moodle, also web 2.0 technologies).

4 Pilot experience: methodology

The described automated activity scheduling tool was deployed in a cybersecurity course, in the context of the Computer Engineering degree at Universidad Politécnica de Madrid, Spain. This subject is scheduled in the third course (second term). The pilot experience was carried out for three years, from 2020 to 2022.

The teaching methodology in this cybersecurity course remained stable for the three years involved in the pilot experience. Additionally, data from the previous year (2019) were collected to be used as a control group. The teaching methodology was also equivalent during the previous year. In general, the entire cybersecurity course is based on the "Learning by Doing" paradigm. Two different methodologies are implemented. First, Challenge-based learning (CBL). And later, Project-based Learning (PBL). The course has a duration of sixteen weeks. During the first ten weeks, short challenges are proposed to develop working competencies independently and at an increasing depth level are proposed. Students must solve the challenge proposing a creative software or procedure to be later publicly presented and explained. Typically, students must solve seven challenges. Later, once students have acquired some cybersecurity abilities, they must develop a project during the six final weeks. In this project, students must build, deploy, and test a cybersecurity infrastructure, usually consisting of a Virtual Private Network (VPN) and secure web services. Four products must be submitted as a result from this project: the self-deployable software, a technical report, a user-friendly report including a manual, and a pentesting report about the proposed infrastructure. Finally, the entire solution must be publicly presented for evaluation. Students could develop all these evaluation activities independently or in small groups (maximum four people). For students working in small groups, all members had the same schedule, so collaboration was not distorted or difficulted. They, as a group, could choose any available schedule in the tool as any other student. Because of the different schedules, collaboration among groups or independent students was not promoted.

These methodologies are combined with theoretical and practical classes, where professors present the basic ideas, explain the reasoning of different techniques and approaches, etc. In these sessions, six basic units are addressed. In addition, five different competencies are acquired with this subject:

- C1: Management and administration of systems, services, and information platforms
- C2: Knowledge, management and application of quality assurance and security technology to information systems
- C3: Identify and analyze the main components in a given problem, in order to solve it according to some criteria in an efficient manner.
- C4: Capacity to express ideas in a clear and effective manner, using written texts and media, and other graphic elements as a support.
- C5: Knowledge and ability to apply the law in the context of the information systems and the security applications, including standards, national laws, and other recommendations.

Table 1 summarizes the course organization and the structure of the competences. The specific way in which the different challenges and project work the competencies

may vary from one year to the other. Depending on the proposed scenarios, technologies, etc., some variations may apply. However, and according to Table 1, globally the same topics are always finally addressed and with similar intensity.

Table 1. Cybersecurity course: structure

Thematic Unit	Competencies	Intensity [1, 3]
Cryptography	C2	1
	C3	3
Firewall	C2	3
Operating systems	C1	3
	C4	2
Hacking	C2	1
	C3	1
	C4	3
	C5	2
Mobile devices	C1	1
	C5	1
Wireless networks	C1	2

Table 2 shows the firstly selected schedules every year (2020–2022), and a comparison to the schedule in 2019 (when the scheduling tool was not available).

Table 2. Selected schedules: week deadlines

Year	Evaluation Activity								
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
2019	2	4	6	8	10	12	14	15	16
2020	3	6	9	10	13	14	14	15	16
2021	3	6	7	8	10	12	13	14	16
2022	3	6	8	10	12	13	14	15	16

The final objective of this experience is to answer some questions about the effectiveness of the proposed automated activity scheduling tool for improving the students' learning and the quality of the entire evaluation process. Three research questions are discussed in this work:

- RQ#1: Does the proposed automated scheduling tool enable students to improve their academic results?
- RQ#2: Does the proposed automated scheduling tool improve student satisfaction and learning?
- RQ#3: Does the proposed automated scheduling tool improve the quality of the evaluation process?

The validation described in this paper was planned, guided, monitored and evaluated by its authors (hereafter experts), who have more than five years of experience in knowledge management, communication software and tools, and data analysis.

The groups in years 2020–2021 were pilot groups, employed to analyze the performance of the proposed automated activity scheduling tools and their impact on student learning. Data from the year 2019 were considered as control group. It was guaranteed that all groups were composed of comparable populations. Table 3 shows the characteristics of the pilot and control groups.

Table 3. Participants' characteristics

Year	Total Number of Students	Mean Age	Standard Deviation in Age	Women Percentage
2019	74	21.8	1.22	39%
2020	83	21.5	0.91	36%
2021	125	21.3	0.84	31%
2022	115	21.4	0.65	32%

All participants were treated anonymously by experts. No personal data related to identification was stored or distributed outside the official platforms. The experience was carried out under the conditions of respect for individual rights and ethical principles that govern research involving humans.

The students were evaluated according to the evaluation rubrics and according to the criteria shown in Table 1. Academic results were collected to be compared and analyzed using statistical mechanisms. Academic results were normalized to remove incompatibilities because of the use of different scales. Furthermore, at the end of each unit, students and professors answered a very short survey with only six short questions using the Likert scale [36] (strongly agree to strongly disagree). The surveys were collected online.

5 Results and discussions

Table 3 shows the normalized academic results of the students in the different pilot groups and the control group. Although academic results may show the first evidence of improvement in the students' learning, a scientific statistical analysis is needed before drawing any final conclusions. We employ a Mann-Whitney U test to perform that analysis. The Mann-Whitney U test is a nonparametric test of the null hypothesis that two samples come from the same population compared to an alternative hypothesis, comparing the mean values of the two samples. It is used to evaluate whether two different data populations are similar or different (higher or lower). The p-value indicates the significance level of the Mann-Whitney U test. Table 4 shows the results obtained from this statistical test as well.

Table 4. Students' academic results

Year	Academic Results			Mann-Whitney U	
	Mean	Std	No-Show (%)	p-value	Significance
2019	0.689	0.244	12%	N/A	N/A
2020	0.817	0.195	8.5%	4.134E-3	**
2021	0.833	0.186	5.9%	4.324E-3	**
2022	0.823	0.188	2.3%	0.975E-3	***

Notes: NS not significant; *significant at $p < 0.05$; **significant at $p < 0.005$; ***significant at $p < 0.001$.

As can be seen, academic results improved round 20% in average during all years involved in the pilot experience. This improvement is stable over time. Additionally, the standard deviation also reduced by around 20%, so the proposed automated scheduling tool also helped to homogenize the level of competence acquisition among all students. Even more important is the reduction in the percentage of students who do not show. During the experience, the number of students that could not able to complete all the assignments and could be evaluated continuously every year. After the implementation of the new scheduling tool, this no-show percentage reduced around 30%. But, after three years, the reduction is slightly higher than 80%. This is probably the most positive impact reported by academic results.

In any case, statistical tests are needed to scientifically conclude if academic results improved thanks to the new automated activity scheduling tool or not. As can be seen from the results of the Mann-Withey U test, a significant improvement is reported in all pilot groups participating in the experience. This improvement is stable in time, and even its significance goes up in the last year studied.

Taking into account all these results, we can answer RQ#1 in a positive way and confirm the proposed new scheduling tool allows students to improve their academic results.

In order to address the two remaining research questions, students and professors responded to a short survey after each year. The survey included six short questions. The questions were equivalent for professors and students, although the subjects in sentences may change since all the questions are student-centered. The questions had to be answered using the Likert scale, where responses ranged from "Totally agree" to "Totally disagree". On this scale, it is possible to assign a numerical value to each response (from five for "Totally agree" to the unit for "Totally disagree"), so statistical tests and mathematical processing are possible. A final section where participants could provide unstructured free comments was also included.

The first four questions (SQ#1, SQ#2, SQ#3 and SQ#4) are related to student satisfaction and learning; the last two questions (SQ#5 and SQ#6) are focused on the quality of the evaluation process. These questions are:

- SQ#1: I [the students] acquired all competencies and abilities planned for this course.
- SQ#2: I feel satisfied with my performance and learning in this course.
- SQ#3: I feel my [the students'] learning and knowledge increased.
- SQ#4: Overall, I am satisfied with the course and the evaluation mechanism and the schedule.

- SQ#5: I [the students] could show and prove all the acquired competencies and abilities during evaluation.
- SQ#6: I [the students] felt comfortable with the evaluation activity schedule.

Table 5 shows the results of the surveys in the different pilot and control groups, as well as the results of the Mann-Withey U test.

Table 5. Students' academic results

Year	**2019**			
Question	**Answers (Surveys)**		**Mann-Whitney U**	
	Mean	**Std**	**p-value**	**Significance**
SQ#1	3.785	2.162	N/A	N/A
SQ#2	3.469	2.489	N/A	N/A
SQ#3	2.575	3.119	N/A	N/A
SQ#4	3.649	2.255	N/A	N/A
SQ#5	2.957	2.959	N/A	N/A
SQ#6	4.142	1.890	N/A	N/A
Year	**2020**			
Question	**Answers (Surveys)**		**Mann-Whitney U**	
	Mean	**Std**	**p-value**	**Significance**
SQ#1	4.792	1.547	3.112E-3	**
SQ#2	4.959	1.814	4.285E-3	**
SQ#3	3.849	2.196	1.656E-3	**
SQ#4	4.757	1.350	6.020E-4	***
SQ#5	3.655	2.547	2.630E-3	**
SQ#6	4.655	1.840	4.541E-3	**
Year	**2021**			
Question	**Answers (Surveys)**		**Mann-Whitney U**	
	Mean	**Std**	**p-value**	**Significance**
SQ#1	4.171	1.917	3.892E-3	**
SQ#2	4.392	1.831	7.482E-4	***
SQ#3	4.035	2.351	4.505E-3	**
SQ#4	4.439	1.616	0.838E-3	***
SQ#5	3.706	2.753	2.290E-2	*
SQ#6	4.097	1.568	3.133E-3	**
Year	**2022**			
Question	**Answers (Surveys)**		**Mann-Whitney U**	
	Mean	**Std**	**p-value**	**Significance**
SQ#1	4.381	2.126	1.524E-3	**
SQ#2	4.032	2.076	8.258E-4	***
SQ#3	3.952	2.531	4.383E-3	**
SQ#4	4.678	1.794	9.961E-5	***
SQ#5	3.795	2.337	0.282E-2	**
SQ#6	4.162	1.779	4.427E-4	***

Notes: NS not significant; *significant at $p < 0.05$; **significant at $p < 0.005$; ***significant at $p < 0.001$.

As can be seen, a significant stable in time improvement is reported for all survey questions, including those related to student satisfaction and the quality of the evaluation process. The mean response to all surveys' questions improved between 25% and 30% after implementing the new proposed scheduling tool. In general, using the new activity scheduling tool, students and professors report that they feel more satisfied with their schedule and learning, they perceive a higher level of learning and competence acquisition, and a higher Quality-of-Experience regarding the evaluation process is also mentioned.

On the other hand, the standard deviation is reduced. This reduction is less relevant than the one observed in the mean responses but still ranges between 5% and 20% depending on the question and year. As concluded when analyzing the academic results, that means that the proposed activity scheduling tool not only increases student learning and their satisfaction, but also homogenizes the level of learning and satisfaction among all students in the pilot groups.

Before answering RQ#2 and RQ#3, a statistical analysis is needed. As can be seen in Table 4, a significant improvement is reported for the question of all surveys. This improvement is especially significant for SQ#4. Also, the significance level is stable in time (and quite elevated) for this question. Thus, students and professors report significantly higher global satisfaction with the learning level and evaluation activity schedule when the proposed tool is used. As a result, we can answer RQ#2 in a positive way.

Finally, although significance is not as high as the one reported for SQ#4, question SQ#6 also shows a relevant and significant improvement. This question refers to the Quality-of-Experience regarding the evaluation process and, mainly, the activity schedule. Therefore, we can conclude that the proposed scheduling tool improves the quality of the evaluation process, and RQ#3 is answered positively. The remaining questions (especially SQ#5), where a significant improvement is also reported, support these conclusions.

A qualitative analysis of the comments provided by participants was also carried out. In general, professors reported an increasing inefficiency in their evaluation activities. As every student has a different deadline for evaluation activities, professors cannot evaluate all activities together. At the beginning, this continuous evaluation process was reported to be more time consuming than other traditional approaches. However, this situation was mitigated as professors found new ways to organize the evaluation process in the second and third year of the experience.

6 Conclusions

In this paper we propose a new automated scheduling tool to fill this gap. The tool is based on the graph theory. The tool employs graph coloring algorithms to calculate all possible schedules for evaluation activities. Students may choose freely the schedule that best fits their personal situation, learning progress, background, etc. using a web portal where all solutions from the coloring algorithm are displayed.

This tool was deployed in the Computer Engineering degree at Universidad Politécnica de Madrid, Spain. A pilot experience was conducted for three years

(2020–2022) in the context of a cybersecurity course to analyze how this new tool improves the learning and the evaluation process of students.

The results show a relevant improvement in the academic results, as well as their satisfaction with the evaluation process. Students also report a higher learning level, a smaller no-show number of students, and a globally better Quality of Experience. All improvements are statistically significant.

The proposed experience was limited to one subject, but it is mature enough to be extended to several courses. However, some challenges should be addressed. First, incompatibilities among evaluation activities are much more difficult to identify when several courses are considered together. Besides, exhaustive coloring algorithms have a n^2 complexity, so processing delays go up quickly. Powerful computing facilities may be needed to support a wider usage of the tool.

7 Acknowledgment

The research leading to these results has received funding from the Ministry of Science, Innovation and Universities through the COGNOS project. (PID2019-105484RB-I00) and from Universidad Politécnica de Madrid through "Competiciones educativas internacionales para promover la mejora de los resultados, la motivación, la acción tutorial y la evaluación continua, en currículos basados en la adquisición de competencia" Project (IE22.6101).

8 References

[1] Mareca, M. P., & Bordel, B. (2019). The educative model is changing: toward a student participative learning framework 3.0—editing Wikipedia in the higher education. Universal Access in the Information Society, 18(3), 689–701. https://doi.org/10.1007/s10209-019-00687-6

[2] Castedo, R., López, L. M., Chiquito, M., Navarro, J., Cabrera, J. D., & Ortega, M. F. (2019). Flipped classroom—comparative case study in engineering higher education. Computer Applications in Engineering Education, 27(1), 206–216. https://doi.org/10.1002/cae.22069

[3] Sumbawati, M. S., & Anistyasari, Y. (2018). The impact of research-based learning on student's academic performance and motivation. In IOP Conference Series: Materials Science and Engineering, (Vol. 296, No. 1, p. 012043). IOP Publishing. https://doi.org/10.1088/1757-899X/296/1/012043

[4] Carlson, L. E., & Sullivan, J. F. (1999). Hands-on engineering: learning by doing in the integrated teaching and learning program. International Journal of Engineering Education, 15(1), 20–31.

[5] Cifrian, E., Andrés, A., Galán, B., & Viguri, J. R. (2020). Integration of different assessment approaches: application to a project-based learning engineering course. Education for Chemical Engineers, 31, 62–75. https://doi.org/10.1016/j.ece.2020.04.006

[6] Félix-Herrán, L. C., Rendon-Nava, A. E., & Nieto Jalil, J. M. (2019). Challenge-based learning: an I-semester for experiential learning in mechatronics engineering. International Journal on Interactive Design and Manufacturing (IJIDeM), 13(4), 1367–1383. https://doi.org/10.1007/s12008-019-00602-6

[7] Phang, F. A., Pusppanathan, J., Nawi, N. D., Zulkifli, N. A., Zulkapri, I., Harun, F. K. C., ... & Sek, T. K. (2021). Integrating drone technology in service learning for engineering students. International Journal of Emerging Technologies in Learning (Online), 16(15), 78. https://doi.org/10.3991/ijet.v16i15.23673

[8] Bordel Sánchez, B., Alcarria Garrido, R. P., & Robles Valladares, T. E. (2019). Industry 4.0 paradigm on teaching and learning engineering. International Journal of Engineering Education, 35(4), 1018–1036.

[9] Alcarria, R., Bordel, B., & de AndrÃ, D. M. (2018). Enhanced peer assessment in MOOC evaluation through assignment and review analysis. International Journal of Emerging Technologies in Learning (iJET), 13(1), 206–219. https://doi.org/10.3991/ijet.v13i01.7461

[10] Bordel, B., Alcarria, R., Robles, T., & Martin, D. (2021). The gender gap in engineering education during the COVID-19 lockdown: a study case. International Journal of Engineering Pedagogy, 11(6). https://doi.org/10.3991/ijep.v11i6.24945

[11] Bordel, B., Alcarria, R., & Robles, T. (2022). A cybersecurity competition to support the autonomous, collaborative, and personalized learning in computer engineering. In 2022 IEEE Global Engineering Education Conference (EDUCON), (pp. 1346–1354). IEEE. https://doi.org/10.1109/EDUCON52537.2022.9766611

[12] Ohland, M., Sill, B. L., & Crockett, E. R. (2002). Thinking about the scheduling of the introduction to engineering syllabus: using a just in time approach. In 2002 Annual Conference, (pp. 7–1203).

[13] Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A review of optimization algorithms for university timetable scheduling. Engineering, Technology & Applied Science Research, 10(6), 6410–6417. https://doi.org/10.48084/etasr.3832

[14] Gashgari, R., Alhashimi, L., Obaid, R., Palaniswamy, T., Aljawi, L., & Alamoudi, A. (2018). A survey on exam scheduling techniques. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1–5). IEEE. https://doi.org/10.1109/CAIS.2018.8441950

[15] Mitsui, H., Sugihara, H., & Koizumi, H. (2004). A scheduling method of distance learning classes including remote experiments. In 18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004. (Vol. 1, pp. 439–444). IEEE.

[16] Mohamad, M. A., Putra, Z. A., Bilad, M. R., Nordin, N. A. H. M., & Wirzal, M. D. H. (2021). An excel based tool development for scheduling optimization. ASEAN Journal of Science and Engineering Education, 1(1), 7–14. https://doi.org/10.17509/ajsee.v1i1.32398

[17] P. Kenekayoro. (2019). Incorporating machine learning to evaluate solutions to the university course timetabling problem. Covenant Journal of Informatics and Communication Technology, 7(2), 18–35.

[18] Daniel, P. G., Maruf, A. O., & Modi, B. (2018). Paperless master timetable scheduling system. International Journal of Applied, 8(2). https://doi.org/10.30845/ijast.v8n2a7

[19] Larabi Marie-Sainte, S. (2015). A survey of particle swarm optimization techniques for solving university examination timetabling problem. Artificial Intelligence Review, 44(4), 537–546. https://doi.org/10.1007/s10462-015-9437-7

[20] Phillips, A. E., Walker, C. G., Ehrgott, M., & Ryan, D. M. (2017). Integer programming for minimal perturbation problems in university course timetabling. Annals of Operations Research, 252(2), 283–304. https://doi.org/10.1007/s10479-015-2094-z

[21] Pillay, N. (2016). A review of hyper-heuristics for educational timetabling. Annals of Operations Research, 239(1), 3–38. https://doi.org/10.1007/s10479-014-1688-1

[22] Mohmad Kahar, M. N., & Kendall, G. (2015). A great deluge algorithm for a real-world examination timetabling problem. Journal of the Operational Research Society, 66(1), 116–133. https://doi.org/10.1057/jors.2012.169

[23] Burdett, R. L., & Kozan, E. (2010). A sequencing approach for creating new train timetables. OR spectrum, 32(1), 163–193. https://doi.org/10.1007/s00291-008-0143-6

[24] Williams, K. M., & Shapiro, T. M. (2018). Academic achievement across the day: evidence from randomized class schedules. Economics of Education Review, 67, 158–170. https://doi.org/10.1016/j.econedurev.2018.10.007

[25] Humphrey, M., & Singh, A. (2017). Reducing class-scheduling conflicts using linear programming. Journal of Professional Issues in Engineering Education and Practice, 143(4), 05017004. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000336

[26] Jha, S. K. (2014). Exam timetabling problem using genetic algorithm. International Journal of Research in Engineering and Technology, 3(5), 649–654. https://doi.org/10.15623/ijret.2014.0305120

[27] Dener, M., & Calp, M. H. (2018). Solving the exam scheduling problems in central exams with genetic algorithms. Mugla Journal of Science and Technology, 4(1), 102–115. https://doi.org/10.22531/muglajsci.423185

[28] Seisarrina, M. L., Cholissodin, I., & Nurwarsito, H. (2018). Invigilator examination scheduling using partial random injection and adaptive time variant genetic algorithm. Journal of Information Technology and Computer Science, 3(2), 113–119. https://doi.org/10.25126/jitecs.20183250

[29] Awad, F. H., Al-Kubaisi, A., & Mahmood, M. (2022). Large-scale timetabling problems with adaptive tabu search. Journal of Intelligent Systems, 31(1), 168–176. https://doi.org/10.1515/jisys-2022-0003

[30] Ahmad, F., Mohammad, Z., Hassan, H., Rose, A. N. M., & Muktar, D. (2015). Quadratic assignment approach for optimization of examination scheduling. Applied Mathematical Sciences, 9(130), 6449–6460. https://doi.org/10.12988/ams.2015.53292

[31] Oyeleye, C. A., Olabiyisi, S. O., Omidiora, E. O., & Oladosu, J. B. (2012). Performance evaluation of simulated annealing and genetic algorithm in solving examination timetabling problem. Scientific Research and Essays, 7(17), 1727–1733. https://doi.org/10.5897/SRE11.2103

[32] Jamaluddin, N. F., & Aizam, N. A. H. (2016). Timetabling communities' demands for an effective examination timetabling using integer linear programming. International Journal of Mathematical and Computational Sciences, 10(5), 263–268.

[33] Akbulut, A., & Yilmaz, G. (2013). University exam scheduling system using graphcoloring algorithm and rfid technology. International Journal of Innovation, Management and Technology, 4(1), 66.

[34] Eley, M. (2006). Ant algorithms for the exam timetabling problem. In International Conference on the Practice and Theory of Automated Timetabling, (pp. 364–382). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77345-0_23

[35] Clear, A., Parrish, A., Zhang, M., & van der Veer, G. C. (2017). Cc2020: a vision on computing curricula. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, (pp. 647–648).

[36] Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert scale: Explored and explained. British Journal of Applied Science & Technology, 7(4), 396. https://doi.org/10.9734/BJAST/2015/14975

[37] Chen, X., Yue, X. G., Li, R., Zhumadillayeva, A., & Liu, R. (2021). Design and application of an improved genetic algorithm to a class scheduling system. International Journal of Emerging Technologies in Learning (iJET), 16(1), 44–59. https://doi.org/10.3991/ijet.v16i01.18225

[38] Wen-jing, W. (2018). Improved adaptive genetic algorithm for course scheduling in colleges and universities. International Journal of Emerging Technologies in Learning, 13(6). https://doi.org/10.3991/ijet.v13i06.8442

9 Authors

Borja Bordel received the B.S. and M.S. degrees in telecommunication engineering from the Technical University of Madrid, in 2012 and 2014, respectively, and the Ph.D. degree in 2018. He is currently an Associate Professor with the Computer Science School. His research interests include cyber-physical systems, wireless sensor networks, radio access technologies, communication protocols, and complex systems. https://orcid.org/0000-0001-7815-5924

Ramón Alcarria received the M.S. and Ph.D. degrees in telecommunication engineering from the Technical University of Madrid, in 2008 and 2013, respectively. He is currently an Associate Professor with the Department of Geospatial Engineering, Technical University of Madrid. He has been involved in several Research and Development European and National projects related to Future Internet, the Internet of Things, and Service Composition. His research interests include service architectures, sensor networks, human-computer interaction, and prosumer environments. E-mail: ramon.alcarria@upm.es, https://orcid.org/0000-0002-1183-9579

Tomás Robles is a full professor at the E.T.S.I Telecommunication of the Technical University of Madrid, UPM. He received a M.S. and Ph.D. degrees in Telecommunication Engineering from Technical University of Madrid in 1987 and 1991 respectively. His research interests are advanced applications and services for broadband networks and technology for engineering education. E-mail: tomas.robles@upm.es, https://orcid.org/0000-0002-6940-8421

Article submitted 2022-08-24. Resubmitted 2022-12-17. Final acceptance 2023-01-14. Final version published as submitted by the authors.