International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 489-498. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2025.v16i3.852

www.editada.org

University Timetabling Problem: An analysis of the state of the art

Marco A. Aguirre Lam [1*], Fausto Balderas[1], Nelson Rangel-Valdez[1], José A. Martinez-Flores[1], Alan G. Aguirre L[1]. and José A. Pérez-Vázquez[1].
[1] División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero, México.
marco.al@cdmadero.tecnm.mx*, fausto.bj@cdmadero.tecnm.mx, nelson.rv@cdmadero.tecnm.mx, jose.mf@cdmadero.tecnm.mx, alan.al@cdmadero.tecnm.mx, g24073003@cdmadero.tecnm.mx

Abstract. The University Course Timetabling Problem is regarded as one of the most significant administrative activities in academic institutions. It is framed as a combinatorial optimisation problem involving the scheduling of courses, students, lecturers, and classrooms, making it extremely complex. This work presents an analytical study of the Timetabling Problem, including its classifications, techniques, advantages, disadvantages, and a case study based on the model implemented at the Instituto Tecnológico de Ciudad Madero.
Keywords: Timetabling Problem, combinatorial optimization.

Article Info
Received February 01, 2025
Accepted March 21, 2025

1 Introduction

Nowadays, the scheduling problem is found in any activity in daily life that involves scheduling. Scheduling consists of deciding which activities will be performed in time slots. The university course timetabling problem is a particular case of combinatorial optimization that consists of planning a school's teaching program as a function of several variables and constraints. This problem is considered NP-Complete and its complexity is relative to the number of variables involved in the model (Even et al, 1975; Garey & Johnson, 1979). One of the main challenges in university course scheduling is to find the best time allocation option in which courses must be assigned according to bounded schedules to satisfy hard and soft constraints, and thus optimize the possible objective functions of the problem. Hard constraints are those that must be enforced. For example, a hard constraint could be that a teacher cannot be in two different courses at the same time. If these constraints are not respected, the schedule will simply not work. On the other hand, soft constraints are more flexible and are desires or preferences that do not necessarily have to be followed. For example, it might be preferable for a teacher not to have classes very early in the morning, but it is not essential. Not complying with these restrictions does not make the schedule impossible, although it might make it less ideal.

Educational Timetabling problems have been studied for decades; however, the Course Timetabling approach has not been developed as quickly as other works, due to the number of variants and the complexity they present (Zhu et al, 2022). There are numerous related works in the literature that address the problem using specific search methods. These works often claim to find the best solution to the problem, yet they employ methods that do not guarantee the optimal result. In other words, the most appropriate method for solving the problem is not always used.

Course Timetabling is a major challenge that impacts both students and teachers. The proper distribution of classes, classrooms and schedules has a significant influence on academic efficiency and educational quality. Incorrect planning can lead to overlapping classes, uneven distribution of workload and conflicts between mandatory and elective subjects, which harms student performance and teacher satisfaction. Adequate organization is essential to maximize the use of institutional resources, such as physical spaces and technological equipment. The solution to this problem not only optimizes academic logistics, but also contributes to a more organized and orderly educational environment, in which all members of the university community can perform better.

This paper is organized as follows. Section 2 presents a brief history of the development of educational timetabling problems. In Section 3, a general classification of university timetabling problems is provided. Section 4 outlines the restrictions associated with these types of problems. Benchmarks relevant to this area are described in Section 5. Section 6 classifies the methodologies used to solve university timetabling problems. In Section 7, various software tools applicable to these problems are discussed. Section 8 presents a case study on the specific model encountered at Instituto Tecnológico de Ciudad Madero. Finally, Section 9 explores the conclusions and future directions of this field.

2 Brief history

The university scheduling problem is a problem that has been studied for more than 60 years and continues to be one of the most researched topics to date due to the various types of problems it presents. Being a problem with many variants, constraints and extremely complex, it has required to be approached from the point of view of computational complexity, applying various types of techniques in order to approximate the best results.

The development of the educational timetabling problem has evolved significantly over the years. The earliest studies on educational timetabling date back to the 1960s. These early generations of systems focused primarily on the constraints and factors identified by the department head in charge of generating the schedules. These systems were very basic and laid the foundation for today's timetabling systems (Dempster et al, 1975).

In 1963, Gotlieb published one of the first papers on the construction of class and teacher schedules, in which he applied a systematic approach, used optimization techniques and algorithms to solve the problem. His work laid the foundation for future research in this field and highlighted the importance of scheduling in educational management (Goh et al, 2020).

In 1965, Csima conducted research on scheduling problems. His work was based on graph theory and combinatorial optimization, and laid the foundation for future research in this field. Csima also addressed the computational complexity of these problems, stressing the importance of developing practical and efficient solutions (Csima, 1965).

Later, in 1971, Werra presented a heuristic algorithm for the construction of school schedules. This algorithm is based on an exact method that applies to a particular family of timetabling problems (Werra, 1971). This was an important advance in the methodology for dealing with this type of problem.

During the 1990s, genetic algorithms and other heuristic techniques were introduced to solve scheduling problems. These techniques made it possible to tackle more complex problems and find more efficient solutions.

From 2000 onwards, research in this field has continued, with the development of new algorithms and approaches to improve the efficiency and effectiveness of scheduling. In addition, benchmarks and data sets have been established to evaluate and compare different methods.

The problem of educational timetabling continues to be an active area of research due to its complexity and its impact on the efficiency of educational institutions.

3 Classification of university timetabling problems

At present, there are many works applied to various scheduling problems, not only in education. Some of these problems are carried out in the transportation sector, hospitals, businesses, and other types of events (Humphreys et al, 2022; Reid et al, 2019; Sterbenk, 2022; Zhang et al, 2024). Scheduling is a complex task in any industry that requires efficient organization.

In the area of University timetabling problems, we can classify this type of work into two main branches:

- **University Course Timetabling (UCTT)** refers to the task of organizing university course schedules in a way that efficiently allocates classes, professors, students and available classrooms. This problem poses a complex optimization challenge because it must take into account multiple constraints and preferences.

- **University Examination Timetabling (UETT)** is a combinatorial optimization problem that focuses on university examination scheduling. The objective is to assign a set of exams to a given number of exam periods, such that strict constraints are met and more flexible constraints are minimized.

In this paper, we will mainly focus on UCTT. For a better understanding of the different classifications of the state of the art, refer to the research papers described by Bashab (2023), and Di Gaspero (2003).

The UCTT problem can be classified into Post-Enrolment Course Timetabling (PE-UCTT), Curriculum-Based Course Timetabling (CB-UCTT) and University Course Timetabling (UCTT) (Arratia-Martinez et al, 2021; Ceschia et al, 2023). Figure 1 shows the classification of these concepts.

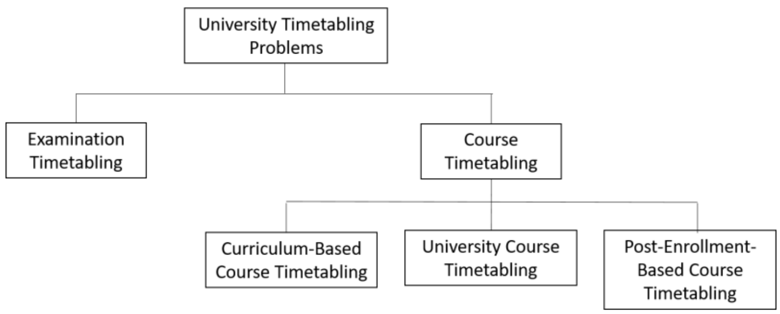

Fig. 1. University Timetabling Problems.

3.1 Curriculum-Based Course Timetabling

The CB-UCTT problem was proposed by Di Gaspero (Di Gaspero, L. & Schaerf, 2003). It is a combinatorial optimization problem that focuses on scheduling university course timetables based on the curriculum. The objective is to find the best weekly allocation of university classes into classrooms and time slots, satisfying a set of hard constraints and taking into account soft constraints. Some of the recent related work will be mentioned below.

In Akkan and Gulcu (2022), the problem is addressed using a bicriteria optimization approach with simulated annealing. The main objective is to find a good approximation to the Pareto frontier by considering two criteria: solution quality and robustness. The study uses instances from ITC-2007 (Bonutti et al, 2012). This approach enables the generation of high-quality schedules that are robust to potential changes in the data, which is crucial in dynamic educational environments. The obtained solutions are evaluated in terms of both quality and robustness, and are compared with other existing techniques to demonstrate the effectiveness of the proposed approach.

The work by Feutrier et al. (2023) focuses on automating the configuration of a university timetable solver. Their research demonstrates that simpler configurations can outperform methods that rely on multiple complex algorithms. The study utilizes the Hybrid Local Search (HLS) method, which combines Hill Climbing, Great Deluge, and Simulated Annealing, along with five different neighborhood operators. By employing automatic configurators such as ParamILS or irace, the researchers are able to discover new, promising configurations without requiring user intervention. This research uses instances from ITC-2007.

In Steiner et al (2024), the problem is addressed by taking into account individual student curricula. Unlike traditional approaches, this work focuses on course scheduling that respects students' individual academic paths, adding a level of complexity to the problem. The main task is performed using intricate integer linear programming. The study uses instances from the university timetabling scheduling problem of the ITC-2019 competition as a reference (Musa & Oyelakin, 2024). The metrics used to evaluate the quality of the schedule include maximizing the number of assigned courses and minimizing constraint violations.

Some of the main benchmarks for real instances of this type of problem have been taken from ITC-2007 and ITC-2019. Additionally, other instances have been sourced from various universities that may have different characteristics, such as those mentioned in (Colajanni & Daniele, 2021).

Several works have aimed to enhance the use of artificial instances for this type of problem. Related studies can be found in (De Coster et al, 2022; Even et al, 1975).

3.2 University Course Timetabling

The task of the UCTT is to organize university course timetabling in a way that efficiently allocates classes, teachers, students and available classrooms (Arratia-Martinez et al, 2021; Müller et al, 2018). This problem was proposed in the ITC-2019 competition. The features of the ITC-2019 instances enable participants to develop and test their algorithms in a controlled yet realistic environment, facilitating the comparison and evaluation of different approaches. Some of the most recent work is mentioned below.

In Ogunkan et al (2024), a graph coloring algorithm utilizing greedy search is proposed, where courses are represented as vertices and the connections between courses that share common students or teachers are represented as edges. The algorithm is tested on a real university instance. The first-fit algorithm facilitates efficient scheduling, minimizing conflicts between courses shared by students or teachers. This algorithm may not be as adaptable as other procedures in circumstances where dynamic adjustments or frequent changes are needed. The evaluation metrics used in this work include program volume, program length, program effort, program difficulty, and execution time.

In Trigos and Coronel (2023), a transdisciplinary framework is introduced that optimizes Expected Academic Performance (EAP) by assigning professors to courses, sections, and schedules that best align with their preferences and profiles. By considering the preferences of teachers, students and other stakeholders, the approach seeks to improve the satisfaction of all. The proposed framework is applied to a real university instance.

In Zhu et al (2022), a conceptual model for the school timetable problem is proposed, taking into account the availability, preferences, and experience of educators. The proposed approach is more efficient than the traditional constraint programming method and provides better solutions. This work applies an Artificial Bee Colony algorithm using artificial instances.

Some of the actual instances used in this kind of problem can be found in the ITC-2019 and UniTime resources[1]. These two organizations co-organized the 2019 International Timetabling Competition. In this competition, research teams compete to build course timetable solvers and solve real-world problems collected from institutions using the UniTime application.

3.3 Post-Enrollment-Based Course Timetabling

The PE-UCTT problem was first proposed in the Metaheuristic Network project, sponsored by the Improving Human Potential program of the European Community[2] coordinated by Prof. Marco Dorigo. The PE-UCTT is a combinatorial optimization problem that focuses on scheduling course timetables after students have already enrolled in courses (Goh et al, 2020). Unlike University Course Timetabling, which is performed prior to enrollment, PE-UCTT faces the additional complexity of adjusting existing schedules to accommodate the preferences and constraints of students and faculty who are already enrolled. Recent work in this area is mentioned below.

In Goh et al (2020), local search techniques (an hybrid of Tabu Search) and an algorithm called Simulated Annealing with Reheating are combined to find feasible solutions and then minimize soft constraint violations, this provides feasible solutions and minimizes soft constraint violations. In the experiments conducted, they used instances from ITC-2002, ITC-2007 and those used in (Socha, 2002) testing different versions of the proposed algorithm.

In Dimitsas et al (2023), a local search procedure complemented with a basic mathematical model and its variations is proposed, offering competitive results in a reasonable execution time, comparable to some of the best-known results. By using exact methods it provides optimal and accurate solutions to the problem, ensuring that all hard constraints are satisfied. The instances used in this work were taken from ITC-2002 and ITC-2007. The results only show the comparison of both benchmarks with respect to their own results when applying their model.

The benchmarks used for this type of problem are primarily sourced from the International Timetabling Competition, which includes both artificial and real instances. In these studies, experimental benchmarking is performed by comparing the results obtained in these competitions with those of each proposed model.

[1] https://www.unitime.org/

[2] https://www.metaheuristics.org/

4 Constraints

Constraints are divided into two main categories: hard constraints and soft constraints (Arbaoui, 2014).

Hard constraints. These constraints must be mandatorily met. If any is violated, the solution is considered invalid. The constraints that have been identified in the various related works are the following: clashing, availability, capacity, conflict number of weeks in the semester and the teaching week.

Some common hard restrictions include the following:

1. Classroom Availability. Classes must be scheduled in classrooms that are available at the designated time.
2. Classroom Capacity: The number of students in a class must not exceed the classroom's capacity.
3. Teacher Availability. Assigned teachers must be available to conduct the scheduled class at the specified time.
4. Scheduling Conflicts. It is important to prevent any student from being scheduled for two classes simultaneously.
5. Course Sequence. Theoretical courses must be scheduled prior to practical courses.

Soft constraints. These constraints are preferences that, if violated, do not invalidate the solution, but make it less ideal. Those found in the literature are the following: teaching space capacity, maximum gap between lectures, time constraints between events, coherence, continuity, clashing, a preferred teaching week, time assignment, spreading events out in time, resource assignment.

Some common soft constraints used in various problems include:

1. Teacher preferences. Assigning courses at times requested by teachers.
2. Preferential use of specialized classrooms or laboratories. Give priority to rooms that have sufficient capacity or special equipment.
3. Gaps between classes. Minimizing gaps or free periods between classes.
4. Student preferences. Assigning courses at times preferred by students.

The constraints depend mainly on the problem being addressed; a more specific classification of the types of constraints used in most related works is presented in (Bashab 2023).

5 Benchmaking

The most commonly used instances for algorithm testing on UCTT problems are based on those from the International Timetabling Competition, which includes the ITC-2002, ITC-2007, and ITC-2019 versions.

The ITC-2002 (First International Timetabling Competition) instances were designed specifically for the competition (McCollum, 2010). The instances include between 50 and 100 exams to be scheduled. Between 20 and 30 classrooms are considered available for scheduling the exams. The instances include constraints such as classroom availability, student schedules, and the need to minimize scheduling conflicts. These instances are designed to represent real scheduling problems, although they are artificial, in order to allow the evaluation of different algorithms and methods.

The ITC-2007 (International Timetabling Competition 2007) instances were divided into three tracks of scheduling problems (Bonutti et al, 2010). The first track focused on timetabling exams at a university, taking into account constraints such as classroom availability, student schedules, and the need to minimize scheduling conflicts. The second track focused on scheduling courses after the initial enrollment period, taking into account student preferences and resource availability. The third track, “Curriculum-based courses,” used instances provided by the University of Udine (Italy). These instances reflected real-world constraints and objectives, such as classroom availability, course capacity, and minimization of scheduling conflicts.

The ITC-2019 (International Timetabling Competition 2019) instances have several features that make them representative of real timetabling problems. The instances include a variety of courses with different time and resource requirements (Müller et al, 2024). They take into account constraints such as classroom availability and student schedules, as well as the need to minimize scheduling conflicts. These include data on students, such as their availability schedule and preferences. Different domains or subject areas are defined for courses, which adds a layer of complexity to scheduling. They also have specific date and time patterns, such as days of the week and times of certain courses. The use of resources such as classrooms and teachers is evaluated to ensure efficient allocation. Weights are assigned to different objectives and constraints to prioritize certain aspects of scheduling. These features allow participants to develop and test their algorithms in a controlled but realistic environment, facilitating comparison and evaluation of different approaches.

6 Methodologies

There are countless algorithms and methods used to solve the UCTT problem. These methods can be classified as follows:

1. Heuristic methods.
2. Metaheuristic methods.
    a. Single solution based approaches.
        i. Simulated annealing.
        ii. Tabu search.
    b. Population based approaches.
        i. Evolutionary algorithms.
        ii. Memetic algorithms.
        iii. Swarm intelligence.
3. Multi-object based approaches.

Some features of the methods mentioned are described below.

Simulated Annealing is a method inspired by the slow cooling process of metals to find approximate optimal solutions. This method is effective in avoiding getting trapped in local optima, allowing it to explore better quality global solutions. The quality of the solutions found may depend on the initial settings of its parameters, such as the initial temperature, the cooling rate, and the number of iterations.

Tabu Search is an efficient method for finding high-quality solutions. It explores the search space more exhaustively than other metaheuristic methods. By using a tabu memory, the algorithm avoids revisiting already explored solutions and can escape local optima. The performance of this algorithm depends on the configuration of parameters, such as the size of the tabu memory and the aspiration criteria.

Evolutionary algorithms are inspired by natural evolution to find optimal or near optimal solutions. They easily adapt to different types of constraints. Using mutation operators to generate a diversity of solutions helps to escape from local optima and allows finding better solutions. In this type of algorithm, the time needed to converge to an optimal solution can be considerable, depending on the complexity of the test instance.

Memetic algorithms blend evolutionary algorithms with local search techniques. They aim to enhance solution quality by merging global exploration with local exploitation. This approach allows for continuous improvement of the solutions discovered.

Multi-objective approaches enable the optimization of multiple objectives simultaneously, resulting in more balanced solutions that meet a variety of needs and constraints. Applying these approaches to scheduling problems allows for a thorough analysis of the trade-offs between different objectives. This analysis helps in selecting solutions that align best with the priorities of the specific problem.

Among the different methods employed, metaheuristics have proven to be the most effective for solving the university timetabling problem. Recently, the use of multi-objective optimization within these methods has become increasingly popular, as it enables the identification of robust and balanced scheduling solutions.

While metaheuristic algorithms can identify satisfactory solutions within a reasonable timeframe, they do not guarantee the discovery of the global optimal solution. Their effectiveness is influenced by the configuration of the parameters used, as well as the size and complexity of the problem instance.

For further information on the methodologies used for this type of problem, refer to the works of Bashab et al. (2023), Abdipoor et al (2023), and Musa & Oyelakin (2024).

7 Tools and applications

There are various software tools currently available to tackle the issue of academic scheduling. These solutions differ in their complexity, functionality, and methodological approach. Below, we present some of the main categories of these tools.

7.1 Comprehensive Academic Management Systems

These platforms offer functionalities to generate timetables, as part of a broader set of educational management tools (Al-Yakoob & Sherali, 2015). Examples include:

- **ASC Horarios.** A specialized solution that uses advanced algorithms for the automatic generation of timetables, considering multiple constraints simultaneously (https://www.asctimetables.com/timetables es.html).
- **Aula1.** It is a comprehensive management software designed for schools, colleges and academies. It facilitates the day-to-day management of these institutions by enabling administrators and teachers to handle tasks such as class timetabling, student management and academic performance tracking (https://www.aula1.com/).
- **BEST (Bullet Education Scheduling and Timetabling**). It is a software platform developed by Bullet Solutions that optimizes the scheduling process of classes, students, classrooms and academic staff for higher education institutions (https://bulletsolutions.com/automated-scheduling-and-timetabling/).
- **DocCF.** It is an educational management platform developed by CF Developer Group. It is designed to help schools manage all school, academic and administrative processes in one place (http://www.grupocfdeveloper.com/plataformapara-colegios.php).
- **FET (Free Timetabling Software).** It is an open source alternative that allows institutional restrictions and preferences to be defined in detail (https://lalescu.ro/liviu/fet/).

7.2 Cloud-Based Applications

The trend towards cloud computing has resulted in solutions that are accessible from any device with an internet connection (Dorneles, 2024). Some applications of this type are the following:

- **GHCHorarios.** It stands out for its intuitive interface and its ability to handle large volumes of data and complex constraints (https://www.penalara.com/es/queesghc).
- **Untis.** It offers advanced optimization functionalities and is widely used in European educational institutions (https://www.untis.at/).

There is no way to define which of the tools presented is better than the others, as there is no scheduling tool that applies to all situations. The choice of the right software depends on the specific needs of the institution. Each of these programs has its own advantages and may be more suitable depending on the size of the institution, the educational level and the specific functionalities required.

8 Modeling a case of study

The Instituto Tecnológico de Ciudad Madero (ITCM) is a public university that represents one of the 254 campuses of the Tecnológico Nacional de México (TecNM) in Mexico. The TecNM presents a particular case of the timetabling problem due to the conditions and restrictions contemplated in the institution's specific guidelines. This university has one of the largest enrollments of the technological system in Mexico and currently the ITCM has 7664 students in 25 educational programs, including undergraduate and graduate degrees. Its current schedule consists of 2,514 courses, of which 762 are shared among educational programs, and the institution has 447 professors assigned to departments and 481 classrooms distributed in three locations in the metropolitan area.

In the particular case of TecNM, the following generalities must be taken into account in the scheduling proposal:

- Academic departments may have one or several academic programs assigned to them.
- Each department has its own core faculty, who may be loaned to other academic departments to teach specific courses, and may teach courses in both departments.
- Each bachelor's degree program has at least two specialties (courses that can be enrolled in after 60% of the approved credits, considering a total of 250 credits per program).

- The minimum academic load is 20 credits and the maximum is 36 credits per semester (one credit is equivalent to one hour of class time).
- If a student has a special course (course not approved for the second time), they can only enroll using the minimum number of credits. If they have taken two special courses, they may only enroll in those two during the corresponding period. If they are unable to enroll in these courses, they must enroll with the minimum load.
- As part of the student's education, students must take complementary activities (worth 5 credits), which begin in the first semester, including 2 tutoring courses (one per semester, a total of 2 credits). From the first semester onwards, sports, academic and cultural activities (worth one credit) may be taken, as long as they do not exceed two credits per semester and the same activity is not repeated during the educational program.
- Courses may have practical hours that can be scheduled in specialized classrooms (laboratories).
- The student must complete a social service program (which has a value of 10 credits) and can be carried out after 70% of approved credits and having completed the 5 complementary credits mentioned above.
- The student must complete an internship program (worth 10 credits) that can be carried out once the student has completed the social service program.
- Complementary activities, social service and internships are considered as courses in the curricular plan.

Students may enroll in the courses that correspond to each semester, according to their curricular plan. The educational programs offer courses called core courses (parent courses) that can be shared with other educational programs, which are called parallel courses (child courses). Each course can have a configuration of theoretical (traditional) or practical (laboratory) hours. The capacity of each course depends on the number of spaces available in each selected classroom.

There are common restrictions in this type of problem, such as, for example, that there can only be one course per classroom, that a single teacher cannot teach more than one course at a time, among others. In the problem presented, the restrictions mentioned above are considered, together with the availability of the teacher's schedule and the limited capacity of the classrooms. One situation that presents the problem encountered at the ITCM is that, given the preferences or considerations of the students, the alternative list that is prepared every semester is never adequately covered, so that students do not take all the courses they should according to their curricular plan and situations arise in which they do not carry the minimum course load per semester. This is not only due to the lack of adequate course planning, but also to the preferences of the students themselves. The soft constraints that are present in this problem are the following:

- Regular courses may be cancelled if there are less than 15 students enrolled and specialty courses may be cancelled if there are less than 10 students enrolled.
- Gaps between courses.
- Student preferences:
    - Scheduling (morning/evening).
    - Instructor.
- Mobility (distance between classrooms):
    - For students.
    - For teachers (due to disability or distance between classrooms and campus).

Considering these constraints further complicates this type of problem, as the mere resolution of conflicts significantly impacts the performance of the algorithms used. Addressing all these constraints necessitates minimizing conflicts in course scheduling, free periods between classes, inefficient use of classrooms, and the geographical dispersion of consecutive courses. Conversely, it requires maximizing the time preferences of teachers and students, ensuring the efficient use of educational resources, balancing the academic workload distribution, and maintaining flexibility to adapt to the changing needs of the problem.

9 Conclusions

The university timetabling problem remains a highly complex challenge, with its complexity escalating in relation to the number of variables and constraints involved in the modeling process. Although numerous studies address these issues, the importance of using real, challenging instances largely depends on the specific considerations included in each research problem. While creating artificial instances that mimic realworld scenarios allows us to analyze algorithm behavior under certain conditions, it does not guarantee that these conditions will actually occur. Many research papers use benchmarks that are not truly challenging. Moreover, several papers that claim good performance do not make their instances publicly available, and most of them utilize ITC instances.

There are numerous tools and papers related to the problem of educational timetabling. However, much of the complexity arises from the challenge of adapting these tools to the specific needs of each institution.

Although this type of problem has been studied for over 60 years, there remains a need to develop new formulations and benchmarks that accurately represent the characteristics of real-world challenges.

This paper analyzes the distinct aspects of contemporary research on the problem of university timetabling, aiming to enhance the understanding of the various characteristics within the current state of the art. Additionally, it introduces a specific instance of this issue occurring at the Technological Institute of Ciudad Madero in Mexico, which will be explored further in subsequent studies.

10 Acknowledgments

The authors are also grateful for the support obtained from: Tecnológico Nacional de México through the Instituto Tecnológico de Ciudad Madero, for the use of the Laboratorio Nacional de Tecnologías de Información (LaNTI), and for the support received through the TecNM project 22557.25-P Design of an intelligent virtual agent for decision support with impact on the navigation of electric vehicles.

References

Abdipoor, S., Yaakob, R., Goh, S., & Abdullah, S. (2023). Meta-heuristic approaches for the university course timetabling problem. *Intelligent Systems with Applications, 19*.

Akkan, K., & Gülcu, A. (2022). Bi-criteria simulated annealing for the curriculum-based course timetabling problem with robustness approximation. *Journal of Scheduling, 25*(4), 477–501.

Al-Yakoob, S., & Sherali, H. (2015). Mathematical models and algorithms for a high school timetabling problem. *Computers & Operations Research, 61*, 56–68.

Arbaoui, T. (2014). *Modeling and solving university timetabling* (Doctoral thesis, Université de Technologie).

Arratia-Martinez, N., Maya-Padron, C., & Avila-Torres, P. (2021). University course timetabling problem with professor assignment. *Mathematical Problems in Engineering*.

Bashab, A., Ibrahim, A., Hashem, I., Aggarwal, K., Mukhlif, F., Ghaleb, F., & Abdelmaboud, A. (2023). Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. *Computers, Materials & Continua, 74*(3), 6461–6484.

Bonutti, A., De Cesco, F., Di Gaspero, L., & Schaerf, A. (2012). Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research, 194*, 59–70.

Ceschia, S., Di Gaspero, L., & Schaerf, A. (2023). Educational timetabling: Problems, benchmarks, and state-of-the-art results. *European Journal of Operational Research, 308*(1), 1–18.

Colajanni, G., & Daniele, P. (2021). A new model for curriculum-based university course timetabling. *Optimization Letters*, 1601–1616.

Csima, J. (1965). *Investigations on a timetable problem* (PhD thesis, University of Toronto, Institute of Computer Science).

De Coster, A., Musliu, N., Schaerf, A., Schoisswohl, J., & Smith-Miles, K. (2022). Algorithm selection and instance space analysis for curriculum-based course timetabling. *Journal of Scheduling, 25*, 35–58.

Dempster, M., Lethbridge, D., & Ulph, A. (1975). School timetabling: A technical history. *Educational Research, 18*(1).

Di Gaspero, L., & Schaerf, A. (2003). Multi-neighbourhood local search with application to course timetabling. In E. Burke & P. D. Causmaecker (Eds.), *Practice and Theory of Automated Timetabling IV* (Vol. 2740, pp. 1–13). Springer.

Dimitsas, A., Nastos, V., Gogos, C., & Valouxis, C. (2023). An exact-based approach for the post-enrolment course timetabling problem. In *Proceedings of the 26th Pan-Hellenic Conference on Informatics (PCI '22)* (pp. 77–82). Association for Computing Machinery.

Dorneles, Á., de Araújo, O., & Buriol, L. (2024). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research, 52*, 29–38.

Even, S., Itai, A., & Shamir, A. (1975). On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing, 5*(4), 691–704.

Feutrier, T., Veerapen, N., & Kessaci, M. (2023). Improving the relevance of artificial instances for curriculum-based course timetabling through feasibility prediction. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (pp. 203–206).

Feutrier, T., Veerapen, N., & Kessaci, M. (2023). When simpler is better: Automated configuration of a university timetabling solver. In *2023 IEEE Congress on Evolutionary Computation (CEC)* (pp. 01–08). IEEE.

Garey, M., & Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. Freeman.

Goh, S., Kendall, G., & Sabar, N. (2020). An effective hybrid local search approach for the post-enrolment course timetabling problem. *OPSEARCH*, 1131–1163.

Gotlieb, C. (1963). The construction of class-teacher timetables. In *Proceedings of the IFIP Congress*.

Humphreys, P., Spratt, B., Tariverdi, M., Burdett, R., Cook, D., Yarlagadda, P., & Corry, P. (2022). An overview of hospital capacity planning and optimisation. *Healthcare, 10*(5), 826.

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., & Burke, E. K. (2010). Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing, 22*, 120–130.

Müller, T., Rudová, H., & Müllerová, Z. (2018). University course timetabling and international timetabling competition 2019. In *Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2018)* (pp. 5–31).

Müller, T., Rudová, H., & Müllerová, Z. (2024). Real-world university course timetabling at the international timetabling competition 2019. *Journal of Scheduling*, 5–31.

Musa, U., & Oyelakin, A. (2024). A survey of approaches for designing course timetable scheduling systems in tertiary institutions. *Journal of Systems Engineering and Information Technology, 3*(1), 1–6.

Ogunkan, S., Idowu, P., Omidiora, E., & Oyeleye, C. (2024). First fit algorithm: A graph colouring approach to conflict-free university course timetabling. *Asian Journal of Research in Computer Science, 17*(5), 125–139.

Reid, K., Li, J., Brownlee, A., Kern, M., & Veerapen, N. (2019). A hybrid metaheuristic approach to a real-world employee scheduling problem. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference*.

Socha, K., Knowles, J., & Sampels, M. (2002). A max–min ant system for the university course timetabling. In M. Dorigo, G. D. Caro, & M. Sampels (Eds.), *Ant Algorithms* (pp. 1–13). Springer.

Steiner, E., Pferschy, U., & Schaerf, A. (2002). Curriculum-based university course timetabling considering individual course of studies. *Central European Journal of Operations Research*.

Sterbenk, Y. (2022). Exploring critical issues in event planning through a group research and jigsaw presentation project. *Journal of Hospitality, Leisure, Sport & Tourism Education*.

Trigos, F., & Coronel, R. (2023). A transdisciplinary approach to course timetabling: An optimal comprehensive campus application. *Product, Management & Development, 21*(1).

Werra, D. (1971). Construction of school timetables by flow methods. *INFOR: Information Systems and Operational Research, 9*(1), 12–22.

Zhang, Q., Lusby, R., Shang, P., Liu, C., & Liu, W. (2024). Solving the train-platforming problem via a two-level Lagrangian relaxation approach. *arXiv preprint arXiv:2401.12345*.

Zhu, K., Li, L. D., & Li, M. (2022). School timetabling optimisation using artificial bee colony algorithm based on a virtual searching space method. *Mathematics, 73*(10).