JOSEIT JOURNAL OF SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY
Published by: Indonesian Informatics Association
e-ISSN:2829-310X

A Survey of Approaches for Designing Course Timetable Scheduling Systems in Tertiary Institutions

Usman Bala Musa[1], Akinyemi Moruff Oyelakin[2*]
[1]Information and Communication Technology Unit, Ibrahim Badamasi Babangida University, Lapai, Nigeria
[2]Department of Computer Science, College of Information and Communication Technology,
Crescent University, Abeokuta, Nigeria

[*]*moruff.oyelakin@cuab.edu.ng*

Abstract. Scheduling the course schedule in tertiary institutions is a complex and crucial task. Past studies have pointed out that when scheduling is performed effectively, it influences students' learning experiences, faculty workloads, and overall institutional efficiency. It has also been argued that in the allocation of courses, classrooms, and faculty members, various constraints, preferences, assumptions, dependencies, and objectives must be taken into consideration. This article reviewed different approaches that have been employed in designing course schedule scheduling systems with particular reference to tertiary institutions. Relevant articles were sourced from notable research repositories using identified keywords. The articles obtained were categorized according to the different methods that were used to solve the scheduling problems of course schedules in higher institutions. The review evaluated how each approach addresses the challenges in course time table scheduling. Thereafter, the paper discussed the advantages, limitations, and suitability of these scheduling techniques time-tabling. Additionally, real-world implementations in various tertiary institutions are mentioned. By discussing the strengths and weaknesses of different methodologies in this work, this survey is believed to be a valuable resource for future studies in the area of course scheduling in tertiary institutions.

Keywords: *Scheduling Problem, Educational Time Table Scheduling, Hard Constraints, Soft Constraints*

Cite as: U. B. Musa and A. M. Oyelakin, “A Survey of Approaches for Designing Course Timetable Scheduling Systems in Tertiary Institutions,” Journal of Systems Engineering and Information Technology, vol. 3, no. 1, pp. 1–6, Mar. 2024. DOI: 10.29207/joseit.v3i1.5609

Received by the Editor: 12-12-2023
Final Revision: 02-24-2024
Published: 03-01-2024

1. Introduction

The time table scheduling problem has been a problem in higher institutions of learning worldwide, as colleges and universities have their faculty members from diverse disciplines who work together to achieve established educational objectives. A common challenge faced by most of these institutions is to design a conflict-free schedule that can be used by lecturers or facilitators [1]. The task of creating optimal lecture schedules in educational institutions involves numerous constraints, such as room availability, faculty preferences, student course requirements, and resource limitations. It has been argued that developing efficient scheduling systems is essential to ensure a conducive learning environment and streamline administrative processes.

Research pointed out that timetabling problems have been studied very widely and the ones in the area of Educational Timetabling Problem (ETP) have been found to be most common [2]. The authors further mentioned that ETP is classified into course time, school time, and examination time. Hence, within a university's scheduling framework, the institution allocates student-enrolled courses and instructor-led sessions to a limited set of resources, encompassing designated time slots and available classes. This procedure comes with a multitude of complexities. To illustrate, in a typical educational setting, there exist multiple student groups that might or might not share the same class times for a given course [3].

Consequently, when scheduling lectures, it is essential to avoid any conflicts between students, lecturers, and lecture venues. This necessity transforms the task of organizing university course scheduling into a laborious and intricate undertaking. University scheduling committees cannot approach this distribution in a haphazard manner, as they are bound to consider numerous decisive factors. A variety of constraints play a pivotal role in guiding the process of creating an effective schedule. These restrictions, which consist of the regulations, policies and preferences of the university, instructors and students, can be classified majorly as hard or soft and in some cases medium in nature. Hard constraints represent absolute guidelines that must not be infringed upon or intersected in any way. On the contrary, soft constraints reflect the desires of the involved stakeholders, which can be overlooked without grave consequences. Meanwhile, medium constraints encompass preferences that are preferable not to be violated [4].

The committee in charge of managing schedules must factor in all of these requirements to arrive at an optimal result. Given the complexity of this challenge, timetable administrators invest a substantial amount of time in search of the optimal solution. However, even with substantial experience, the solution they uncover may not be optimal due to the extensive array of potential combinations. Therefore, the equitable distribution of responsibilities among various educators within an academic institution presents a challenge of combinatorial nature. In a broader context, the resolution of such dilemmas and the attainment of precise optimal solutions pose computational intractability.

Thus, the university timetabling predicament exemplifies a hard non-polynomial problem (NP) [5], characterized by the absence of efficient solutions. This complexity is further compounded in timetabling scenarios, where a one-size-fits-all algorithm is unfeasible due to the distinctive constraints of each institution [6]. Additionally, when performed manually, the outcome depends both on the initial approach and the experience of the timetabling committee. In particular, both private and public universities often invest days in manually allocating classes to time slots based on lecturer and lecture venue availability [7]. Hence, the proposition of automating this process is driven by the objective of meeting genuine needs, focusing primarily on reducing the time required to complete while maintaining resource efficiency.

The committee in charge of managing schedules must factor in all of these requirements to arrive at an optimal result. Given the complexity of this challenge, timetable administrators invest a substantial amount of time in search of the optimal solution. However, even with substantial experience, the solution they uncover may not be optimal due to the extensive array of potential combinations. Therefore, the equitable distribution of responsibilities among various educators within an academic institution presents a challenge of combinatorial nature. In a broader context, the resolution of such dilemmas and the attainment of precise optimal solutions pose computational intractability.

In terms of complexity, the process of developing the schedule itself is classified as NP-Complete, while optimizing the schedule is classified as NP-Hard [8]. An NP problem is characterized as one that can be solved by a non-deterministic Turing machine in polynomial time. An NP-complete problem, denoted as X, is one that falls within NP and can be transformed into X in polynomial time. On the other hand, an NP-Hard problem, denoted Y, can be transformed into Y in polynomial time but is not necessarily within NP. In essence, this means that solving the problem in polynomial time is not feasible unless the conjecture P = NP holds.

These types of problem require substantial computational resources for resolution, often requiring the assistance of computers and algorithms, as they typically involve vast and intricate solution spaces [9]. The scheduling of course schedules in tertiary institutions presents a multifaceted challenge that requires the allocation of courses, classrooms, and faculty members to create efficient and effective schedules. This scheduling process must consider a wide array of constraints, preferences, and objectives unique to the context of higher education. However, the complexity of these requirements often leads to suboptimal schedules, which can result in inefficiencies, imbalances of faculty workload, and reduced learning experiences for students.

The core problem addressed in this study revolves around the design and implementation of course timetable scheduling systems for tertiary institutions. Specifically, the study aims to investigate the various approaches used to tackle the following critical challenges, such as the following: Faculty preferences and availability, course conflicts, room allocation, student needs, curriculum constraints, resource utilization, Changing constraints. Research has established that there are many popular approaches used in the study of the lecture timetabling problem in university courses [10], [11]. Some of approaches for attending to time-table scheduling issues include: Operational research-based techniques and metaheuristic approaches.

2. Methods

The methodology used in this paper focuses on reporting different work on designing course timetable scheduling systems for tertiary institutions. The approach follows a methodical approach to identify, choose, evaluate, and combine relevant academic works. Various keywords were used to search for relevant literature across several research databases, such as IEEE, Web of Science, Springer Link, and Science Direct. Thereafter, in this article some of the relevant articles were surveyed.

2.1 Search Strategy

Various iterations of a keyword-driven exploration were conducted employing diverse variations and amalgamations of terms such as the key terms in the research topic "A Survey of Approaches for Designing Course Timetable Scheduling Systems for Tertiary Institutions" include: Survey or review, approaches, system design, course schedule, scheduling systems, and tertiary institutions. These terms are essential for conducting a comprehensive literature search and for structuring research on the various methodologies and strategies used in designing course schedule scheduling systems in tertiary educational settings.

2.2 Study Selection

The objective of this stage is to refine the collection of papers obtained in the initial search, isolating studies with thematic relevance that could potentially address the primary research inquiries. Some set of inclusion and exclusion criteria was established, together with some guidelines to evaluate the quality of the studies. The selection and refinement phases encompassed in this review are outlined below.

a. use the predefined criteria for including and excluding papers;
b. eliminate any duplicated articles that are discovered across various databases;
c. identify all approaches and group papers accordingly;
d. Explore further related articles by referencing the sources cited in the articles acquired from step 3, and subsequently reapply the assessment process as outlined in step 3 to these additional articles.

3. Results and Discussion

The scheduling of course, timetables in tertiary institutions is a complex task that impacts various stakeholders and institutional operations. The literature reveals a growing interest in developing efficient and effective scheduling systems to address the unique challenges faced in this context. This review of the literature synthesizes key findings from existing research, categorizing, and discussing the diverse approaches used to design course timetable scheduling systems specifically tailored for higher institutions.

3.1 Operational Research (OR) Based Techniques

The approach of graph coloring as outlined in Cassar, Titus, and Grill (2017), which modeled the University Course Timetable Problem (UCTP) using an undirected graph [12]. In this approach, vertices represent events, colors signify time-slots, and constraints are depicted as edges within the graph. The primary aim is to devise a timetable in which adjacent events do not share the same color, thus preventing conflicts. Similarly, Hussain, Muhammad and Nawaz (2018) focused on reducing the chromatic number of the graph by segregating graph vertices, similar to the separation of students within courses [13]. This separation strategy improves the quality of the schedule and mitigates penalties compared to manually constructed timetables.

Furthermore, this graph-colouring technique has been extended to schedule classrooms. In this context, the vertices correspond to common courses, while the edges represent students. The target objective is to employ a heuristic approach that achieves two goals as presented by Posada, Andersson and Hall (2017). The goals include:

a. ensuring an even distribution of courses across colors, and;
b. Balance the number of courses assigned to each timeslot, taking into account available rooms.

Another approach combines genetic coloring, as mentioned in Ganguli et al. (2017), to minimize the number of colors needed to color the graph. Kenekayoro et al. (2019) equally introduced integer programming (IP) to address the UCTP, with the goal of directing courses to teachers, student groups, and weekly/daily time periods. Furthermore, Malikov et al. (2018) introduced a two-step relaxation method based on IP to generate efficient timetables [14], [15]. In Step 1, courses requiring consecutive scheduling were allocated to specific days and times, while Step 2 ensured the consecutive scheduling of courses for identical student groups that span multiple time periods. Additionally, Abayomi-Alli, et al (2019) used the Integer Linear Optimization Compiler (ILOC) software to implement a Constraint Satisfaction Problem (CSP) approach for constructing university timetables, with a focus on satisfying event constraints when allocating resources [16].

The third operational research-based methods are the constraint programming methods, which have emerged as powerful tools for handling the various constraints present in course timetable scheduling. These methods involve modeling scheduling constraints as logical relationships, allowing systematic exploration of feasible schedules. Constraint programming techniques have been effective in managing various requirements, including course conflicts, room allocation, and faculty preferences. However, the formulation of accurate constraint models can be challenging. The constraint satisfaction programming (CSP) method is a computational system that defines constraints as limitations within a facilities space. Its primary objective is to identify a consistent set of values, each of which can be assigned to variables while adhering to predefined constraints. This problem is characterized by three variables: CSP = (X, D, C), where X represents a finite set of variables (X = x1, x2, ..., xn), D is a finite set of domain values (D = d1, d2, ..., dn) from which the variables are selected, and C is a finite set of constraints (C = c1, c2, ..., cm) that pertain to specific subsets of variables. The ultimate solution involves assigning values to each variable in a way that satisfies all provided constraints [17].

Moreover, researchers introduced a combination of genetic algorithms with constraint-based reasoning, offering a viable and nearly optimal solution to the course timetabling problem. In a related context, constraint-based reasoning was utilized in an object-oriented approach to address timetabling planning problems.

Furthermore, a specific software was applied to implement the CSP approach for university timetabling [18]. This software was employed to achieve an objective function by ensuring the satisfaction of event constraints during the allocation of resources.

3.2 Meta-Heuristic Approaches

In the domain of metaheuristic methods, one approach is Case-Based Reasoning (CBR), which is a problem-solving approach that leverages solutions from similar previous cases stored in a case base to solve new problems [19]. Retrieval and matching algorithms are employed to locate a source case in the case base that closely resembles the new problem. The solution from the matching source case is used directly or adapted for the new problem, assuming that similar problems have similar solutions [20].

However, there are instances where a particular heuristic that proved effective for solving a specific problem type may not yield favorable results when applied to different problem types [21]. CBR has been applied as a foundational technique for implementing selection constructive hyperheuristics in solving combinatorial optimization problems [22]. Algorithm 1 outlines the process of implementing a constructive hyperheuristic for selection of CBR.

Algorithm 1: Implementing a CBR Selection Constructive Hyper-Heuristic

a. Create an initial case base (see Algorithm 2).
b. Define a similarity measure to calculate the resemblance between cases.
c. Enhance the features and their weights used in the similarity measure through an evaluation of the case based on a training set.
d. Refine the set of cases by evaluating the performance of the CBR system in a training set [20].

The initial case base, as created in Algorithm 2, comprises source cases, each describing a problem state and the most effective heuristic(s) for that source case [21]. Source cases involve problem features, although they can vary in complexity. For example, in the context of the examination timetabling problem, the characteristics may encompass the number of hard constraints, soft constraints, examinations, and the density of the conflict matrix. A heuristic is associated with each source case description, which aids in constructing a solution. In certain studies, the five best performing heuristics are stored for each source case, ranked by objective value.

Algorithm 2: Creating an Initial Case Base

a. Select an initial set of features.
b. Determine the weights (w_i) for each feature.
c. Choose various states of the problem with different characteristics.
d. Solve these problems using different construction heuristics.
e. Store problem states as cases, represented by problem features and the corresponding best-performing construction heuristic(s) [20].

Selecting the appropriate set of features and source cases is crucial to building an effective CBR system. Algorithm 3 starts with an initial set of features, which are subsequently refined to enhance their effectiveness in constructing solutions. These initial features often include all possible characteristics of the state of the problem. In some studies, features are classified as simple, complex, or combinations thereof. To find a solution for a new problem, the system retrieves a source case that most closely matches the new problem case based on a similarity measure determined by the case features. The nearest-neighbor similarity measure is commonly employed [21].

The similarity is calculated as a weighted sum of feature pairs between the source case (SC) and the new problem case (P), with higher values indicating greater similarity. The process of refining the source cases (line 4 of Algorithm 1) involves improving the performance of the CBR system based on an initial case base and similarity measure. Training cases are labeled with the best heuristics obtained through optimization methods. The feature weights in the similarity measure are iteratively adjusted based on these training cases until the retrieved cases closely match the specified heuristics in the training cases [20].

Refinement of source cases also involves retaining only relevant and useful cases that contribute to accurate recommendations of the best heuristics. Various techniques, such as the "Leave-One-Out" method, have been used for system training based on a set of training cases [20]. A drawback of case-based reasoning lies in the reliance of the hyperheuristic's success on the stored problems, potentially limiting its ability to generalize. For example, when dealing with a problem that lacks sufficient similarity to any of the stored cases, finding a solution can be challenging or even impossible [23].

A study conducted that delved into the influence of neighboring structures on the effectiveness of the Tabu search algorithm in tackling the UCTP. This investigation included an examination of the impact of simple and swap transitions on Tabu search operations, drawing on various neighboring structures [24]. Additionally, the study introduced and assessed four novel neighboring structures for evaluation and comparison purposes. In their efforts to tackle the UCTP, this study came up with an approach that combines Kempe neighboring chains within a simulated annealing algorithm [23]. This innovative two-phase strategy involved the initial phase of generating a feasible solution using a heuristic-based graph. In the subsequent phase, a simulated annealing algorithm was employed to improve solution quality by minimizing soft constraint violations. This improvement was achieved through the incorporation of a Kempe-neighboring chain-based heuristic.

3.3 Findings

This work reviewed studies that have proposed different approaches for scheduling course schedules in higher institutions of learning. The findings of the review demonstrated the diversity of scheduling approaches, each with its strengths and limitations. Some of the methods were found to excel in optimizing specific constraints; others offer adaptability to accommodate dynamic changes in the scheduling environment. The review evaluated how each approach addresses the challenges in course time table scheduling. This paper also discussed the advantages, limitations, and suitability of the different scheduling techniques for course time allocation.

In addition, real-world implementations in various tertiary institutions are mentioned. By mentioning the strengths and weaknesses of different methodologies, it is believed that this review can serve as a valuable resource for academic administrators, policymakers, and researchers seeking to enhance the efficiency and effectiveness of course scheduling in tertiary institutions. This study also found that some of the techniques for the design of a University Course Timetabling system are of different categories. By discussing the strengths and weaknesses of different methodologies, it is believed that this review can serve as a valuable resource for future studies seeking to improve the efficiency and effectiveness of course scheduling in tertiary institutions.

4. Conclusions

In this work, a review of various approaches for course schedule scheduling was conducted. The various methods were categorized according to their application to the University Course Timetabling Problem (UCTTP). The investigation revealed that UCTTP involves the intricate task of scheduling school hours to accommodate a series of meetings between teachers and students within a specified time frame while adhering to numerous constraints. These constraints tend to vary between different educational institutions, leading to the development of diverse solutions for this problem. UCTTP is recognized as a combinatorial optimization problem characterized by an extensive search space and a typically high number of constraints. Time table scheduling is generally classified as NP-hard, indicating its computational complexity. This study further confirmed that the field of university course timetabling remains an active and vital area of research, as evidenced by the substantial body of literature devoted to it. The reports in this article have also established those timetabling problems exhibit distinctive characteristics, and there is no one-size-fits-all solution. Each method discussed in this paper addresses a specific scenario or a limited set of instances within the UCTTP domain. In the near future, an improved course schedule scheduling approach will be proposed that can handle scenarios in a real-life higher education institution.

References

[1] Al-Yakoob, Sherali, A. Et, Al, M. [10], and L. Zhang, “A Survey on the Requirements of University Course Timetabling,” *International Science Index, Mathematical and Computational Sciences*, vol. 10, no. 5, 2016.

[2] K. Zhu, L. D. Li, and M. Li, “A Survey of Computational Intelligence in Educational Timetabling,” *Int J Mach Learn Comput*, vol. 11, no. 1, pp. 40–47, Jan. 2021, doi: 10.18178/ijmlc.2021.11.1.1012.

[3] PremasirilDM, “University Timetable Scheduling Using Genetic Algorithm Approach Case Study: Rajarata University OF Sri Lanka,” *Journal of Engineering Research and Application www.ijera.com*, vol. 8, no. 12, 2018.

[4] E. Andrew Okonji, “Intelligence Classification of the Timetable Problem: A Memetic Approach,” *International Journal on Data Science and Technology*, vol. 3, no. 2, 2017, doi: 10.11648/j.ijdst.20170302.12.

[5] R. R. Iwańkowicz and M. Taraska, “Self-classification of assembly database using evolutionary method,” *Assembly Automation*, vol. 38, no. 3, 2018, doi: 10.1108/AA-06-2017-071.

[6] W. Tian, W. Guo, and M. He, “On the Classification of NP Complete Problems and Their Duality Feature,” *International Journal of Computer Science and Information Technology*, vol. 10, no. 1, 2018, doi: 10.5121/ijcsit.2018.10106.

[7] A. E. Phillips, C. G. Walker, M. Ehrgott, and D. M. Ryan, “Integer programming for minimal perturbation problems in university course timetabling,” *Ann Oper Res*, vol. 252, no. 2, 2017, doi: 10.1007/s10479-015-2094-z.

[8] R. Ganguli and S. Roy, “A Study on Course Timetable Scheduling using Graph Coloring Approach,” 2017.

[9] A. Soni, K. Padayachee, and R. Ajoodha, “A Graph Colouring Solution to the Timetable Problem with Hard Constraints in Higher Education Institutions,” *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4332718.

[10] H. Babaei and A. Hadidi, “A Review of Distributed Multi-Agent Systems Approach to Solve University Course Timetabling Problem,” *ACSIJ Advances in Computer Science: an International Journal*, vol. 3, no. 5, 2014.

[11] H. Babaei, J. Karimpour, and A. Hadidi, “A survey of approaches for university course timetabling problem,” *Comput Ind Eng*, vol. 86, 2015, doi: 10.1016/j.cie.2014.11.010.

[12] I. R. Cassar, N. D. Titus, and W. M. Grill, “An improved genetic algorithm for designing optimal temporal patterns of neural stimulation,” *J Neural Eng*, vol. 14, no. 6, 2017, doi: 10.1088/1741-2552/aa8270.

[13] A. Hussain, Y. Shad Muhammad, and A. Nawaz, “Optimization through Genetic Algorithm with a New and Efficient Crossover Operator,” 2018.

[14] F. Ben Amor, T. Loukil, and I. Boujelben, “The new formulation for the Integrated Dial-a-Ride Problem with Timetabled fixed route service,” in *International Colloquium on Logistics and Supply Chain Management, LOGISTIQUA 2019*, 2019. doi: 10.1109/LOGISTIQUA.2019.8907255.

[15] M. Andrey, V. Voronkin, P. Svetlana, and S. Alexey, “Software and hardware infrastructure for timetables scheduling in university,” in *CEUR Workshop Proceedings*, 2018.

[16] O. Abayomi-Alli, A. Abayomi-Alli, S. Misra, R. Damasevicius, and R. Maskeliunas, “Automatic Examination Timetable Scheduling Using Particle Swarm Optimization and Local Search Algorithm,” in *Data, Engineering and Applications: Volume 1*, vol. 1, 2019. doi: 10.1007/978-981-13-6347-4_11.

[17] J. H. Obit and D. Landa-Silva, “Computational study of non-linear great deluge for university course timetabling,” *Studies in Computational Intelligence*, vol. 299, 2010, doi: 10.1007/978-3-642-13428-9_14.

[18] Z. Lixi and L. SimKim, “Constructing university timetable using constraint satisfaction programming approach,” in *Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet*, 2005. doi: 10.1109/cimca.2005.1631445.

[19] A. Agnar and E. Plaza, “Case-Based reasoning: Foundational issues, methodological variations, and system approaches,” *AI Communications*, vol. 7, no. 1, 1994, doi: 10.3233/AIC-1994-7104.

[20] N. Pillay and R. Qu, “Hyper-Heuristics: Theory and applications,” in *Natural Computing Series*, 2018.

[21] E. K. Burke, S. Petrovic, and R. Qu, “Case-based heuristic selection for timetabling problems,” *Journal of Scheduling*, vol. 9, no. 2, 2006, doi: 10.1007/s10951-006-6775-y.

[22] A. Agnar and E. Plaza, “Case-Based reasoning: Foundational issues, methodological variations, and system approaches,” *AI Communications*, vol. 7, no. 1, 1994, doi: 10.3233/AIC-1994-7104.

[23] N. Pillay, “A review of hyper-heuristics for educational timetabling,” *Ann Oper Res*, vol. 239, no. 1, 2016, doi: 10.1007/s10479-014-1688-1.

[24] K. S. Abdallah, “Multi-objective Optimization for Exam Scheduling to Enhance the Educational Service Performance.,” *Journal of Management & Engineering Integration*, vol. 9, no. 1, 2016.