POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)

URL: http://dergipark.org.tr/politeknik

A web-based decision support system for managing course timetabling in online education

Çevrimiçi eğitimde ders çizelgelemesini yönetmek için web tabanlı bir karar destek sistemi

Yazar(lar) (Author(s)): Mevlüt UYSAL[1], Onur CERAN[2], Mustafa TANRIVERDİ[3], Erdal ÖZDOĞAN[4], Mutlu Tahsin ÜSTÜNDAĞ[5]

ORCID[1]: 0000-0002-6934-4421

ORCID[2]: 0000-0003-2147-0506

ORCID[3]: 0000-0003-3710-4965

ORCID[4]: 0000-0002-3339-0493

ORCID[5]: 0000-0001-6198-2819

To cite to this article: Uysal M., Ceran O., Tanrıverdi M.,Özdoğan E. ve Üstündağ M. T., “A web-based decision support system for managing course timetabling in online education”, ***Journal of Polytechnic***, *(*): *, (*).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Uysal M., Ceran O., Tanrıverdi M.,Özdoğan E. ve Üstündağ M. T., “A web-based decision support system for managing course timetabling in online education”, ***Politeknik Dergisi***, *(*): *, (*).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1517479

A Web-based Decision Support System for Managing Course Timetabling in Online Education

Çevrimiçi Eğitimde Ders Çizelgelemesini Yönetmek İçin Web Tabanlı Bir Karar Destek Sistemi

Highlights

- *Developed a web-based DSS using simulated annealing to optimize online course timetabling.*
- *Implemented a neighborhood mechanism for faster algorithm convergence.*
- *Integrated DSS with SIS and LMS for seamless data synchronization and timetable management.*
- *Achieved significant reduction in peak connections, improving bandwidth efficiency.*
- *Enhanced online learning experience with balanced load distribution and minimized server overloads.*

Graphical Abstract

This paper presents a web-based Decision Support System (DSS) using a simulated annealing algorithm to optimize online course timetabling. Integrated with the university's SIS and LMS, the DSS balances server loads and improves bandwidth efficiency, enhancing the online learning experience.

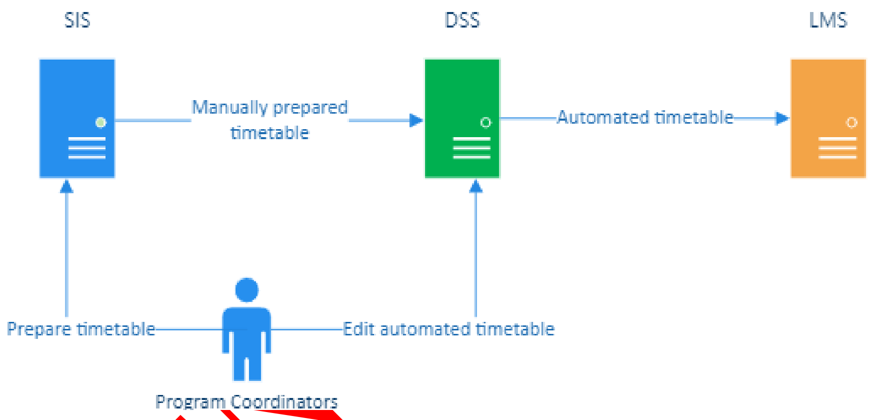

Figure. Integration Between Information Systems

Aim

This study aims to develop and implement a web-based Decision Support System (DSS) to optimize online course timetabling, ensuring balanced server loads and efficient bandwidth usage.

Design & Methodology

The DSS was designed to integrate with the university's SIS and LMS, utilizing a simulated annealing algorithm with a neighborhood mechanism to optimize course timetabling. The system allows user interaction and adjustments, ensuring flexibility and real-time data synchronization.

Originality

This study introduces a novel web-based DSS that leverages a simulated annealing algorithm and a neighborhood mechanism for efficient online course timetabling, integrating seamlessly with existing SIS and LMS systems.

Findings

The DSS significantly reduced peak connections to under 4,000 per time slot, lowered the standard deviation of connections, and achieved a more balanced load distribution compared to manually generated timetables.

Conclusion

The DSS effectively optimized online course timetabling, balanced server loads, and improved bandwidth efficiency, offering a scalable solution for future online education needs and enhancing the overall learning experience.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

A Web-based Decision Support System for Managing Course Timetabling in Online Education

Araştırma Makalesi/Research Article

Mevlüt UYSAL[1*], Onur CERAN[2], Mustafa TANRIVERDİ[1], Erdal ÖZDOĞAN[2], Mutlu Tahsin ÜSTÜNDAĞ[3]

[1]Management Information Systems, Faculty of Applied Science, Gazi University, Ankara, Türkiye

[2]IT Department, Gazi University, Ankara, Türkiye

[3]Department of Computer Education and Instructional Technologies Gazi University, Ankara, Türkiye

(Geliş/Received : 17.07.2024 ; Kabul/Accepted : 30.09.2024 ; Erken Görünüm/Early View : 08.11.2024)

ABSTRACT

The COVID-19 pandemic precipitated an abrupt transition from traditional face-to-face instruction to online learning, posing significant challenges in managing course timetabling and ensuring efficient bandwidth utilization. This paper presents the development and implementation of a web-based Decision Support System (DSS) that employs a simulated annealing algorithm to optimize course scheduling in an online education context. Seamlessly integrated with the university's Student Information System (SIS) and Learning Management System (LMS), the DSS enables automated timetable generation and real-time data synchronization. Program coordinators can make necessary adjustments, while students and instructors access their schedules through a user-friendly interface. Experimental results demonstrate a substantial improvement in the distribution of concurrent connections compared to manually generated timetables, significantly reducing peak server loads by up to 66% and standard deviations. The proposed DSS addresses the immediate challenges of the shift to online education while offering a scalable solution for future needs, thereby enhancing the online learning experience for both students and instructors.

Keywords: Course timetabling, online education, decision support system, simulated annealing algorithm.

Çevrimiçi Eğitimde Ders Çizelgelemesini Yönetmek İçin Web Tabanlı Bir Karar Destek Sistemi

ÖZ

COVID-19 pandemisi, geleneksel yüz yüze eğitimden çevrimiçi öğrenmeye ani bir geçişi zorunlu kılmış ve ders çizelgeleme yönetimi ile verimli bant genişliği kullanımını sağlama konusunda önemli zorluklar ortaya çıkarmıştır. Bu makale, çevrimiçi eğitim bağlamında ders programlamayı optimize etmek için tavlama benzetimi algoritmasını kullanan web tabanlı bir Karar Destek Sistemi'nin (KDS) geliştirilmesini ve uygulanmasını sunmaktadır. Üniversitenin Öğrenci Bilgi Sistemi (ÖBS) ve Öğrenim Yönetim Sistemi (ÖYS) ile sorunsuz bir şekilde entegre olan KDS, otomatik ders programı oluşturma ve gerçek zamanlı veri senkronizasyonu sağlamaktadır. Program koordinatörleri gerekli düzenlemeleri yapabilirken, öğrenciler ve öğretim üyeleri kullanıcı dostu bir arayüz aracılığıyla ders programlarına erişebilmektedir. Deneysel sonuçlar, manuel olarak oluşturulan programlara kıyasla eşzamanlı bağlantıların dağılımında önemli bir iyileşme olduğunu, maksimum sunucu yüklerinin %66'ya varan oranda azaldığını ve standart sapmaların önemli ölçüde düştüğünü göstermektedir. Önerilen KDS, çevrimiçi eğitime geçişin getirdiği acil zorlukları ele almanın yanı sıra gelecekteki ihtiyaçlar için ölçeklenebilir bir çözüm sunarak hem öğrenciler hem de öğretim üyeleri için çevrimiçi öğrenme deneyimini iyileştirmektedir.

Anahtar Kelimeler: ders çizelgeleme, çevrimiçi eğitim, karar destek sistemi, tavlama benzetimi algoritması.

1. INTRODUCTION

The COVID-19 pandemic has significantly disrupted higher education worldwide, necessitating an abrupt shift from traditional face-to-face instruction to online learning modalities. This sudden transition caught many institutions unprepared, leading to numerous challenges, including the rapid adaptation to new teaching methodologies and technological infrastructures while striving to maintain educational quality under constrained circumstances [1-3]. Among the critical issues arising from this transition were challenges related to course scheduling [4-6] and efficient bandwidth management [7-10].

Course timetabling is inherently a complex and time-consuming process for educational institutions, particularly universities. It is characterized as an NP-hard problem involving the assignment of courses to limited time slots and resources—such as instructors and virtual classrooms—while satisfying a variety of constraints. Traditionally, this problem entails assigning courses to time slots and physical classrooms while adhering to hard constraints (e.g., avoiding conflicts for students and instructors) and optimizing soft constraints (e.g., accommodating preferred teaching times and minimizing gaps in student schedules) [11,12]. These constraints are shaped by institutional policies, resource availability, and the preferences of instructors and students.

With the shift to online education during the COVID-19 pandemic, the timetabling problem requires a different approach due to altered constraints. While physical classrooms and their associated limitations are

**Sorumlu Yazar (Corresponding Author)*

e-posta : mevlutuysal@gazi.edu.tr

eliminated, it becomes imperative to distribute the number of simultaneous classes evenly throughout the day to ensure the efficient operation of online education systems. Failure to do so may result in connection problems, audiovisual disruptions, and class cancellations due to system overload.

The course timetabling problem has been extensively studied due to its complexity and practical importance in educational settings. Various methodologies have been proposed to address this problem, ranging from exact algorithms to metaheuristic approaches [13,14]. Metaheuristic algorithms such as simulated annealing, genetic algorithms, and tabu search have been widely applied to tackle the NP-hard nature of timetabling problems, providing near-optimal solutions within reasonable computational times [15,16]. For instance, Mirhassani and Habibi [17] examined timetabling challenges in hybrid education models, while Bellio et al. [18] focused on feature-based tuning of simulated annealing for curriculum-based course timetabling. Akbulut et al. [19] developed a simulated annealing algorithm to address a faculty-level university course timetabling problem with complex constraints, achieving significant improvements over traditional methods. Xiang et al. [20] proposed a two-stage metaheuristic algorithm combining genetic algorithms and enhanced tabu search to tackle the university course scheduling problem with additional constraints on temporal coherence and equitable course dispersion. Similarly, Romeguera et al. [21] developed a web-based course timetabling system using an enhanced genetic algorithm with heuristic mutation, optimizing classroom resources and satisfying both hard and soft constraints. Additionally, the adaptive large neighborhood search algorithm has demonstrated effectiveness in solving complex timetabling problems by efficiently exploring large solution spaces [22].

The literature has explored various facets of the timetabling problem under different constraints and settings. Researchers have employed sophisticated models to address the unique demands of academic institutions. Rappos et al. [23] introduced a mixed-integer programming model for university timetabling, achieving second place in the International Timetabling Competition 2019 with a two-stage optimization method. Mokhtari et al. [24] developed a multi-objective model for postgraduate courses, minimizing scheduling conflicts using the ε-constraint method. Colajanni and Daniele [25] focused on curriculum-based timetabling, optimizing both hard and soft constraints, and applied their model to the University of Catania. Daskalaki et al. [26] presented a two-stage relaxation procedure to efficiently solve timetabling problems using integer programming, significantly reducing computation time. Lindahl et al. [27] explored strategic timetabling by formulating bi-objective models to analyze the impact of resources on scheduling quality. Bagger et al. [28] proposed an integer programming relaxation for weekly course assignments, improving lower bounds and proving optimal solutions for most instances.

The development of Decision Support Systems (DSS) for timetabling has gained considerable attention, with systems like SlotManager [29] and udpSkeduler [30] offering automated solutions for schedule generation. These systems utilize various optimization models to enhance the scheduling process, reduce manual effort, and improve the quality of timetables. Siddiqui et al. [31] discuss a web-based group DSS developed for the Academic Term Preparation problem at a large Middle Eastern university's business school. This system integrates a multi-objective mixed-integer programming model to automate and optimize timetabling, considering curriculum requirements, student sectioning, and institutional policies. Furthermore, DSS are widely employed in solving various other optimization problems [32-34].

The COVID-19 pandemic has introduced new dimensions to the timetabling problem. Studies have addressed the need for hybrid models combining online and face-to-face instruction, considering factors such as reduced classroom capacities and social distancing guidelines [35]. For example, Şimşek [36] investigated an online education setting and proposed a multi-objective mathematical model to balance course distribution and manage bandwidth effectively. Cardonha et al. [37] introduced a DSS developed at the University of Connecticut for Fall 2020, using mixed-integer programming to reassign courses to different teaching modalities and rooms in response to COVID-19 safety standards that drastically reduced room capacities.

Our work contributes to this evolving field by focusing on the unique challenges of online education during the pandemic, specifically aiming to distribute student connections evenly and prevent technical issues caused by bandwidth limitations. Our university experienced significant bandwidth issues due to a large number of simultaneous connections (over approximately 10,000) during peak hours. To address this, we developed a web-based DSS that employs a simulated annealing algorithm to optimize course timetabling.

The DSS integrates seamlessly with the university's Student Information System (SIS) via web services, enabling the automated transfer of comprehensive data on courses, student enrollments, instructors, and existing schedules into the DSS database. This integration ensures that the timetabling process is grounded in real-time information, accurately reflecting the current state of course offerings and enrollment patterns.

Administrative users can configure various parameters through the system interface, such as the number of days per week classes will be held, the start and end times of classes in the morning and evening, and the maximum allowable number of concurrent connections. The system then optimizes the schedule, considering the real-time load balance on Learning Management System (LMS) servers to distribute courses evenly throughout the week.

This dynamic adjustment is crucial for maintaining balanced server loads, preventing bandwidth bottlenecks, and enhancing the overall reliability of online course delivery.

In summary, our research addresses the critical need for an optimized course timetabling system in the context of online education. By leveraging a simulated annealing algorithm within a DSS framework, we provide a scalable solution that balances course schedules, mitigates bandwidth issues, and enhances the online learning experience for both students and instructors.

2. METHOD

To address the complex challenge of course timetabling in an online education environment, we developed an advanced DSS integrated with our university's existing SIS and LMS. The DSS leverages a simulated annealing algorithm to optimize the distribution of courses, ensuring a balanced load on the university's servers and minimizing bandwidth issues.

Our approach involves several key steps:

1. Data Collection and Integration: We gathered comprehensive data on courses, enrollments, instructors, and initial timetables from the SIS. This data was synchronized with the DSS to provide a robust foundation for generating optimized timetables.

2. Algorithm Selection: Given the NP-hard nature of the timetabling problem, we selected a simulated annealing algorithm for its effectiveness in finding near-optimal solutions within a reasonable timeframe. The flexibility of simulated annealing allows it to escape local optima and explore a wide range of potential solutions.

3. System Design and Implementation: We designed the DSS to be user-friendly and interactive, allowing program coordinators to adjust parameters such as class start and end times, maximum concurrent connections, and instructor preferences. This flexibility ensures that the system can meet specific departmental needs while adhering to overall optimization goals.

4. Experimental Setup and Evaluation: We conducted extensive experiments to evaluate the performance of our algorithm, comparing the automatically generated timetables to manually created ones. Metrics such as the number of concurrent connections, standard deviation of connections across time slots, and overall system performance were used to assess the effectiveness of the DSS.

In the following sections, we provide a detailed problem description, outline the specific constraints and requirements of our timetabling problem, and describe the simulated annealing algorithm in detail. We then present our experimental results, highlighting the significant improvements achieved by our system in balancing course schedules and reducing bandwidth issues.

2.1. Problem Description

The COVID-19 pandemic necessitated the delivery of courses entirely or partially through online education platforms. A significant challenge emerged as specific time slots, particularly early morning hours were overly preferred, creating substantial bandwidth strain on the LMS during peak times. This led to disruptions, disconnects, and even cancellations of virtual classes, especially problematic in our university with over 40,000 students.

The solution involves distributing courses evenly throughout the week and ensuring that the number of concurrent connections does not exceed server capacity. Factors such as the days courses are held, start and end times, and the number of time slots per day significantly impact timetable efficiency. We developed an interactive DSS to address these challenges.

Data from the SIS included schedules for traditional education, encompassing 20 academic units and 296 departments, totaling 7,417 unique courses with 21,796 assigned course hours and over 200,000 enrollment records. Due to online education's nature, course durations and weekly contact hours were reduced, resulting in 8,892 course hours after reorganization.

To achieve a balanced distribution of courses, we aimed to minimize the difference between the maximum and minimum number of connections in each time slot per day [36,38]. The timetabling problem incorporates various hard (mandatory) and soft (flexible) constraints [13]:

- All courses in the timetable must be assigned.
- An instructor or student cannot attend multiple classes in a single time slot.
- Course sessions with multiple hours must be scheduled in consecutive time slots.
- The interactive DSS allows parameters such as the start and end times of classes, the number of days courses are held, and the depth of the search space to be set through the system interface. Additionally, the following constraints can be specified as hard or soft:
- Maximum number of concurrent connections.
- Whether the instructor's preferred days should be considered.
- Furthermore, the following constraints can be set as soft through the system:
- Courses should preferably be scheduled according to the instructor's preferred days and times.
- Courses should preferably not be scheduled in time slots after a specified hour.
- Courses should preferably not be scheduled on specified days.

The flexible nature of the DSS allows administrators to create the most suitable timetable for the institution, considering load balancing and reasonable runtimes.

2.2. Simulated Annealing Algorithm

We employed a simulated annealing algorithm (SA) to address the challenge of reducing the disparity in the number of simultaneous connections at various times of the day. The course timetabling problem is NP-hard, and given the vast dataset we are working with, achieving the optimal solution or even a near-optimal solution within a reasonable timeframe is highly improbable. Our primary goal, therefore, is to minimize the difference in the number of concurrent connections as much as possible, ensuring it stays within the maximum capacity that the servers can handle. This approach is sufficient for our problem, given the constraints.

SA is a meta-heuristic algorithm particularly well-suited for this task due to its simplicity in implementation, flexibility in parameter settings, and ability to escape local optima. Traditional local search techniques are inadequate for our needs because they often get trapped in local optima, failing to find an acceptable solution within a practical timeframe [39]. Therefore, simulated annealing, with a guided search mechanism, was chosen for its effectiveness in exploring the solution space more broadly.

The core concept of SA is inspired by the annealing process in metallurgy, where a material is heated and then slowly cooled to remove defects and achieve a more stable structure. In our algorithm, this process is mirrored by starting with a high "temperature" that allows for significant changes in the timetable, followed by a gradual reduction in temperature, leading to more minor and refined adjustments.

Here is a high-level overview of the SA process used in our DSS:

1. Initialization: The algorithm begins with an initial timetable configuration, manually generated by program coordinators.

2. Temperature Schedule: An initial temperature is set, which is progressively decreased according to a cooling schedule. The temperature controls the probability of accepting worse solutions, allowing the algorithm to escape local optima.

3. Neighbor Solution Generation: Instead of choosing entirely random neighbor solutions, the algorithm directs the search towards solutions that potentially have a better average number of connections. Reducing randomness helps focus the search on more promising areas of the solution space.

4. Cost Function: The cost of the new solution is calculated based on the imbalance and deviation costs. Imbalance cost measures the sum of the difference between the maximum and minimum number of connections per time slot for each day, while deviation cost accounts for deviations from preferred days and times.

5. Acceptance Criteria: A new solution is accepted if it improves the current solution. If it does not, it may still be accepted with a probability that decreases with the temperature and the magnitude of the solution's worsening. This probabilistic acceptance helps the algorithm avoid getting stuck in local optima.

```
currentSolution = LoadInitialSolution()
bestSolution = currentSolution
currentCost = CalculateCost(currentSolution)
bestCost = currentCost
initialTemperature, coolingRate, iter
temperature = initialTemperature
WHILE temperature > 1 DO
    iterationsPerTemperature = ComputeIterationsPerTemperature(temperature, initialTemperature, iter)
    newSolution = GenerateNeighborSolution(currentSolution, iterationsPerTemperature)
    newCost = CalculateCost(newSolution)
    IF ShouldAcceptSolution(currentCost, newCost, temperature) THEN
        currentSolution = newSolution
        currentCost = newCost
        IF currentCost < bestCost THEN
            bestSolution = currentSolution
            bestCost = currentCost
        END IF
    END IF
    temperature = temperature * coolingRate
END WHILE
```

Figure 1. The pseudocode of the developed SA algorithm

6. Iteration and Cooling: The process iterates, generating and evaluating neighbor solutions and gradually reducing the temperature. As the temperature decreases, the algorithm becomes less likely to accept worse solutions, honing in on a more refined timetable.

7. Termination: The algorithm terminates after a set number of iterations or when the temperature reaches a minimum threshold, yielding the best solution found.

The pseudocode of the developed SA algorithm is shown in Figure 1.

By employing SA, we can navigate the complex solution space of the timetabling problem more effectively than traditional local search methods. The algorithm's flexibility allows it to adapt to our online education platform's specific constraints and requirements, ensuring a balanced distribution of courses and minimizing bandwidth issues.

2.2.1. Neighbor solution generation

In our SA algorithm, the generation of neighbor solutions is a crucial step. Given the large search space of the timetabling problem, making only a few changes to time slots at each temperature level is not practical. To navigate the search space more efficiently and find better solutions rapidly, we implemented an additional loop allowing multiple changes per temperature iteration. The number of iterations per temperature can be adjusted through the DSS interface, enabling users to balance between solution quality and algorithm runtime. The number of iterations per temperature decreases as the temperature lowers with a ratio of temperature/initial temperature. When the temperature is high, more iterations are performed, allowing for broader solution space exploration. As the temperature decreases, fewer iterations are performed, focusing the search on fine-tuning the solution.

are the detailed steps for generating a new neighbor solution:

Steps to Generate a New Solution

Step 1: Calculate Average Enrollments Per Day

Step 2: Identify Above-Average and Below-Average Time Slots

Step 3: Perform Iterations to Adjust Time Slots

For a specified number of iterations per temperature (configurable through the DSS interface):

- Randomly select a course from the above-average time slots.
- Optionally select a new random day for the course.
- Attempt to move the course to a randomly selected below-average time slot for the selected day.
- Ensure the new time slot is available and does not conflict with existing assignments.

Step 4: Set the New Solution

By incorporating multiple changes in each temperature iteration, the algorithm can explore the solution space more effectively, rapidly moving towards better solutions. Users can adjust the number of iterations per temperature through the DSS interface to achieve a balance between solution quality and runtime, enhancing the flexibility and adaptability of the system. As the temperature lowers, the number of iterations per temperature decreases, focusing the search on fine-tuning the solution.

2.3. Experimental Results

To evaluate the performance of our SA algorithm, we conducted several experiments with carefully chosen parameters. After initial pretests, we set the parameters: an initial temperature of 100, a cooling rate of 0.99, and a maximum of 2000 iterations. We enforced only hard constraints for these tests without altering the instructors' preferred days.

The experiments were performed on a system with an Intel(R) Xeon(R) W-2145 CPU and 32 GB of RAM. Our dataset included 8,892 course hours and 234,828 total enrollments. The application runtime was 3.6 minutes.

We compared the distribution of total connections by time slots between a manually generated timetable and the timetable produced by our SA algorithm.

Table 1 shows the distribution of total connections for the manually generated timetable. There are 14 time slots in a day shown as TS. Some time slots experienced over 10,000 connections, leading to significant bandwidth issues. The standard deviation (Std. Dev.) values across different days were exceedingly high, indicating a substantial imbalance in the distribution of connections.

In stark contrast, Table 2 presents the results from the timetable generated by the SA algorithm. The maximum number of connections in any time slot was significantly reduced to less than 4,000. Moreover, the standard deviation values were markedly lower compared to the manually generated timetable, indicating a much more balanced distribution of connections.

Table 1. Distribution of Total Connections by Time Slots (TS) for Manually Generated Timetable

TS	Days				
	1	2	3	4	5
1	7112	7136	7432	7007	6392
2	7127	7304	7093	8077	6622
3	5646	6968	5558	6161	3889
4	1649	1540	1849	1265	933
5	1802	2003	2105	1045	991
6	10695	11188	11559	11153	8135
7	2957	3949	2857	3183	2829
8	5221	5005	4965	3338	2674
9	1036	1390	1498	1230	897
10	2827	2776	2249	1746	1519

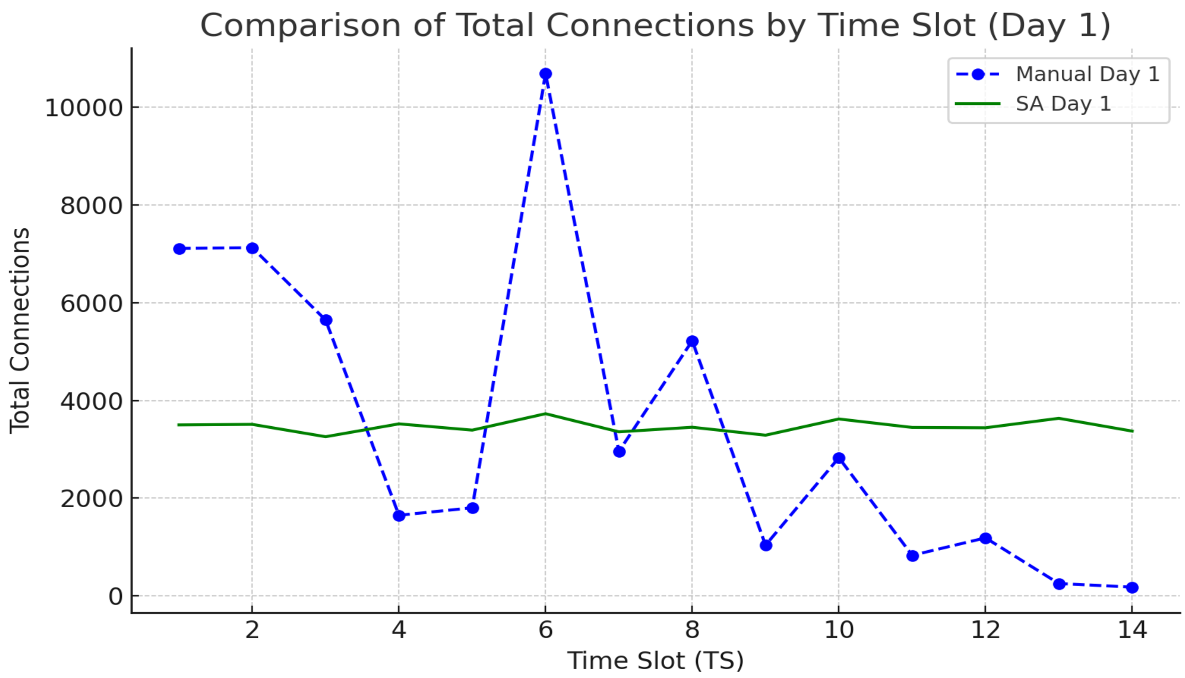

Figure 2. Comparison of Total Connections by Time Slot for Day 1

Table 1.(Cont.) Distribution of Total Connections by Time Slots (TS) for Manually Generated Timetable

11	828	1354	1096	1094	1200
12	1186	1806	1218	1076	515
13	251	223	228	72	140
14	178	161	261	174	185
Mean	3465.36	3771.64	3569.14	3330.07	2637.21
Std. Dev.	3195.36	3289.59	3310.53	3423.33	2638.76

Table 2. Distribution of Total Connections by Time Slots(TS) for Automatically Generated Timetable

Table 2.(Cont.) Distribution of Total Connections by Time Slots(TS) for Automatically Generated Timetable

TS	Days				
	1	2	3	4	5
1	3499	3577	3918	3193	2637
2	3510	3708	3499	3296	2641
3	3257	3586	3332	3228	2633
4	3520	3862	3795	3338	2638
5	3392	3876	3488	3424	2635
6	3728	3897	3652	3308	2638
7	3357	3918	3388	3393	2634
8	3451	3749	3422	3424	2642
9	3288	3787	3608	3478	2636
10	3620	3724	3853	3284	2642
11	3447	3797	3373	3333	2634
12	3440	3832	3373	3252	2644
13	3634	3648	3551	3231	2632
14	3372	3842	3716	3439	2635
Mean	3465.36	3771.64	3569.14	3330.07	2637.21
Std. Dev.	132.82	110.99	192.69	89.58	3.77

The experimental results clearly demonstrate the effectiveness of the SA algorithm in generating a balanced timetable. The manually generated timetable had significant peak connection numbers, with some time slots reaching as high as 11,559 connections, leading to severe bandwidth issues and high variability. The SA algorithm, on the other hand, reduced the peak connections to 3,918, representing an approximate 66% reduction. Furthermore, the standard deviation between the manually generated timetable and the SA algorithm's timetable showed a substantial difference, with the SA algorithm reducing the standard deviation by an average of around 95%, indicating a more balanced load distribution and significantly improved scheduling.

Figure 2 comparing Day 1 of both timetables further illustrates this improvement. The manually generated timetable exhibits sharp peaks, particularly at Time Slot 6, where the connections exceed 10,000. In contrast, the SA-generated timetable shows a much flatter and more consistent distribution of connections, with no slot surpassing 4,000. This visualization highlights how the SA algorithm effectively mitigates peak loads and balances the scheduling more evenly across the day.

These results underscore the utility of the SA algorithm in optimizing the timetabling process for online education, ensuring that bandwidth is efficiently utilized and reducing the likelihood of server overloads. This balanced distribution of connections improves the reliability and quality of the online education experience for both students and instructors.

2.4. Implementation

The DSS is seamlessly integrated with the university's SIS and LMS, facilitating a streamlined process for managing course timetabling. Figure 3 shows the integration between information systems.

The following steps outline the interaction between these systems and the various users involved:

1. Timetable Preparation in SIS: Program coordinators prepare the initial timetable using the SIS. This includes scheduling courses and assigning instructors and students.

2. Synchronization with DSS: The courses, students, instructors, and timetables are synchronized with the DSS via web services. This integration ensures the DSS has the most up-to-date information for generating the timetable.

3. Automated Timetable Generation: The DSS uses the synchronized data to generate an automated timetable. This timetable is optimized to balance the number of connections and adhere to constraints.

4. Adjustments by Program Coordinators: Program coordinators can make minor adjustments to the automatically generated timetable directly within the DSS. This flexibility allows for fine-tuning to meet specific departmental needs.

5. Viewing Timetables: Both students and instructors can view their respective timetables on the DSS. This ensures that everyone is aware of their schedules and can plan accordingly.

6. Synchronization with LMS: Finally, the completed timetable is synchronized with the LMS. This integration ensures that the schedules are reflected in the learning management system, allowing for the smooth execution of online courses.

This integrated system ensures that the timetabling process is efficient, accurate, and user-friendly, enhancing our university's overall management of online education.

3. DECISION SUPPORT SYSTEM

The DSS streamlines the course timetabling process through a flexible, user-friendly interface built using ASP.NET MVC in C# programming language and a Microsoft SQL Server database. It is designed for flexibility, allowing minor adjustments to automatically generated schedules. Integrated user accounts, synchronized with SIS credentials, ensure that each user has appropriate access and permissions based on their role within the university. Program coordinators can modify course schedules for their departments while adhering to overall constraints. Academic unit coordinators oversee and approve all unit course schedules to ensure alignment with institutional policies and objectives. Instructors and students can view their schedules through the system, with instructors seeing their teaching assignments and students viewing their enrolled courses, including any updates made by coordinators.

Users log in to the DSS using their SIS credentials, ensuring secure and seamless access. Upon logging in, users are prompted to select their roles from Academician, Program Coordinator, Academic Unit

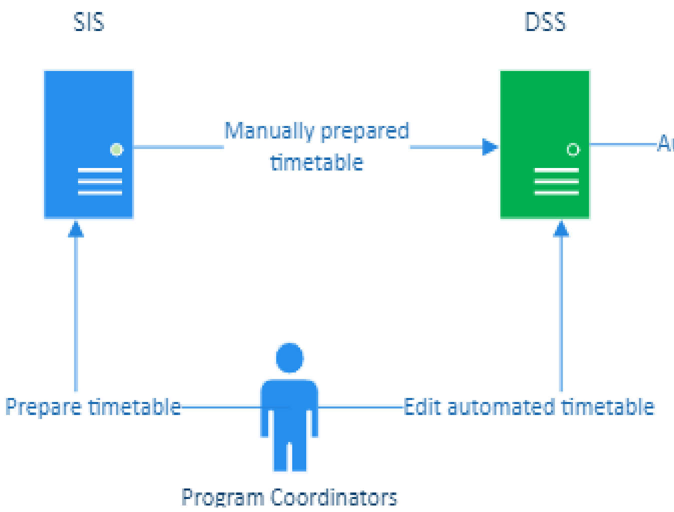

Figure 3. Integration Between Information Systems

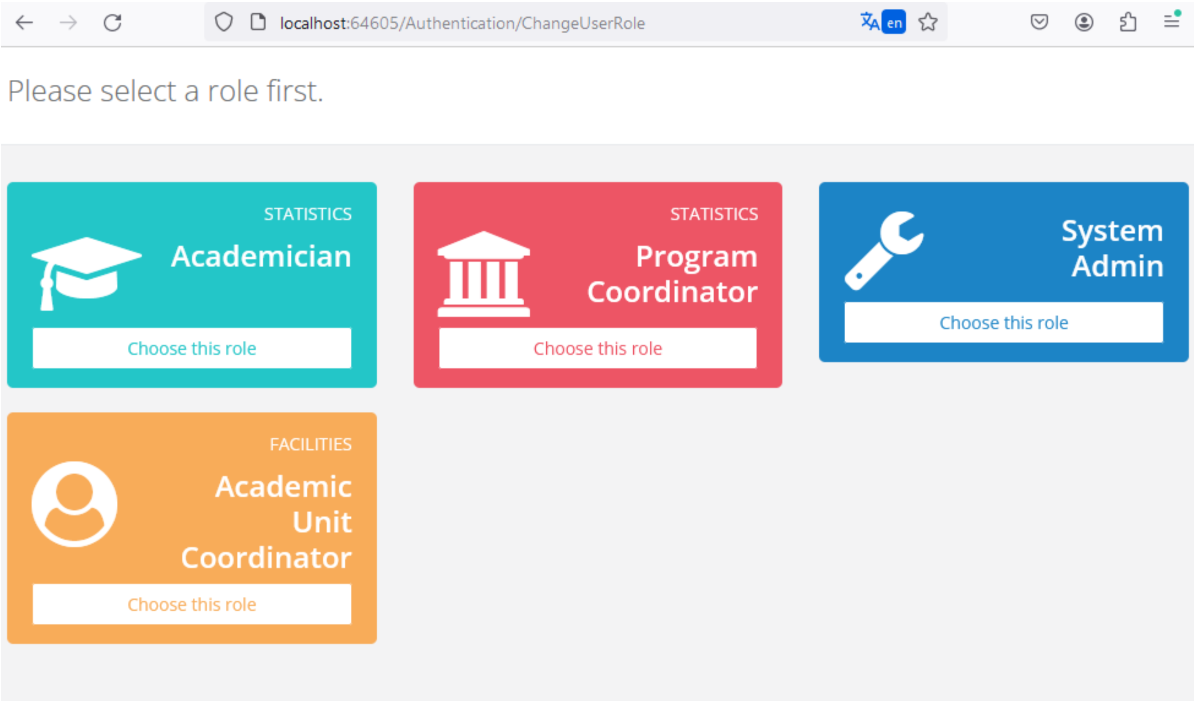

Figure 4. Role selection screen

Coordinator, and System Admin (see Figure 4). Each role has specific permissions and access levels, ensuring users can only perform tasks relevant to their roles. For instance, only System Admins have the privilege to create new timetables, whereas Program Coordinators can make slight edits to the course times.

Once logged in, System Admins can begin creating a new timetable. The timetable creation interface allows admins to configure various parameters to ensure an optimal schedule. The parameters include defining each day's start and end times, limiting the maximum number of simultaneous connections to avoid bandwidth issues, and adjusting the search depth (selecting a value between 1-5) to balance between runtime and solution quality. Additionally, admins can specify preferences such as minimizing the number of classes after a specific time, locking lesson days to prevent changes, considering instructors' preferred times, and selecting the days on which classes should be held. Figure 5 shows the different parameters of the timetable-creating screen.

Once the initial timetable is created, program coordinators can review it and make necessary adjustments. They have the flexibility to fine-tune the schedule to fit departmental needs better. The editing interface allows coordinators to change course times while adhering to the constraints set during the initial timetable creation. This ensures adjustments do not conflict with the overall schedule or exceed the system's capabilities.

As seen in Figure 6, the DSS also visualizes the load distribution daily. This feature is essential for understanding and managing the distribution of courses and their impact on the system. The Course Load Graphs display the number of connections throughout the day, helping administrators to identify peak times and make data-driven decisions to optimize the schedule further.

As seen in Figure 7, program coordinators can change the automatically created timetable according to the instructor's preference. Students and instructors can view their schedules on the DSS, ensuring everyone knows their timetables and can plan accordingly. This feature is crucial for maintaining transparency and ensuring all parties are informed about their schedules. By logging in with their SIS credentials, users can access their personalized schedules, which include all the courses they are enrolled in or teaching.

The DSS is built to handle the complexities of university timetabling efficiently, ensuring both flexibility and control over the scheduling process. By leveraging ASP.NET MVC and Microsoft SQL Server, the system ensures seamless data integration, real-time updates, and secure access for all users. The automated scheduling system, powered by simulated annealing algorithm, optimizes course distribution while accounting for institutional constraints, bandwidth limitations, and user preferences. This combination of automation, flexibility, and user input makes the DSS an essential tool for managing the dynamic and evolving needs of online education, ultimately improving the experience for students, instructors, and administrators alike.

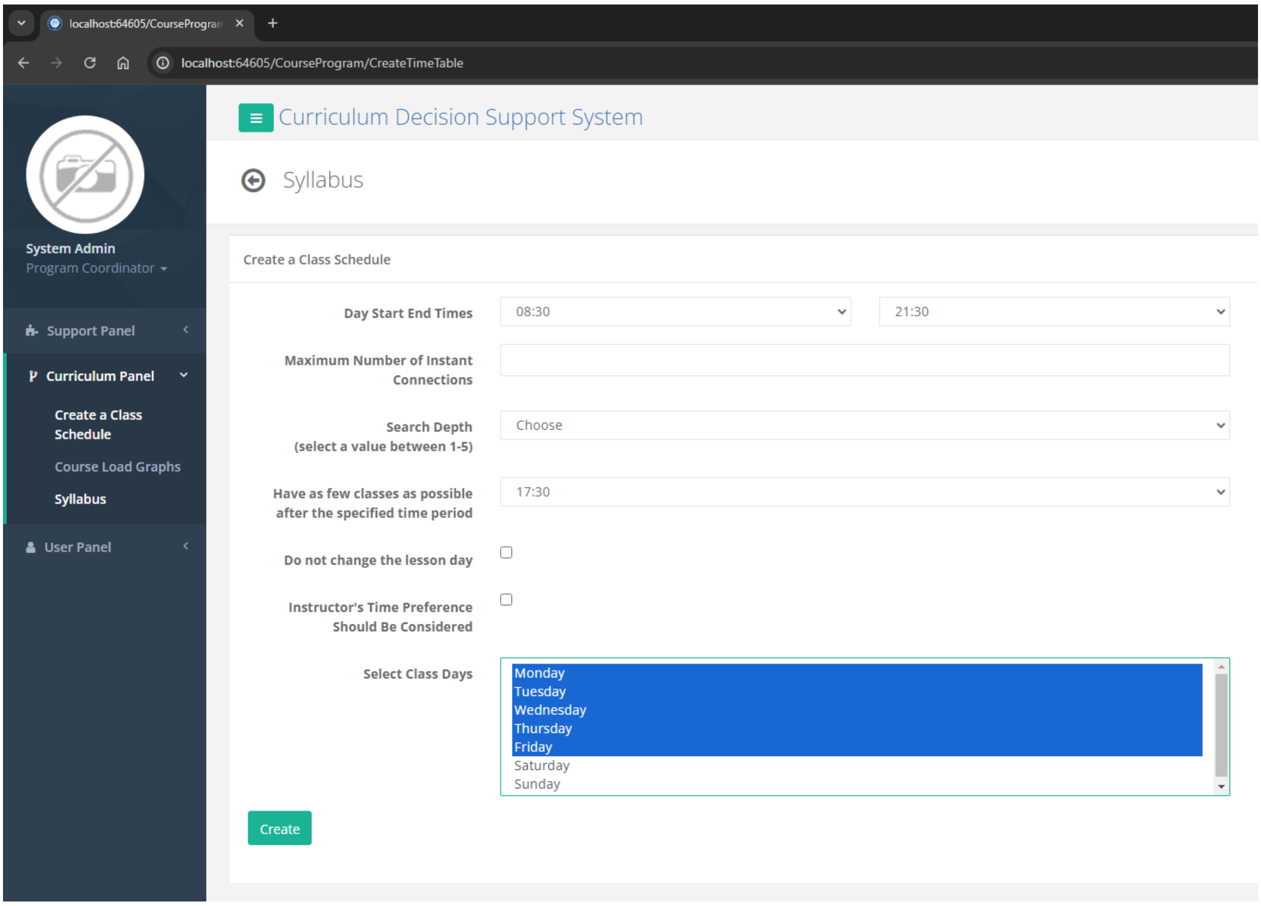

Figure 5. Creating new timetable

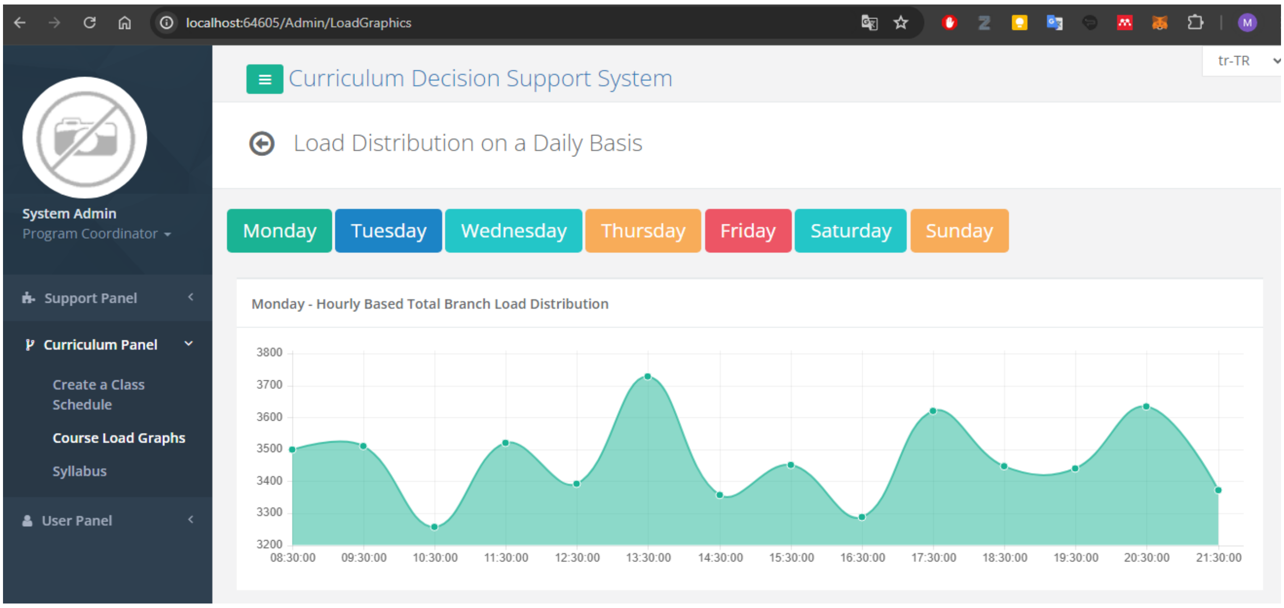

Figure 6. Visualization of the load distribution

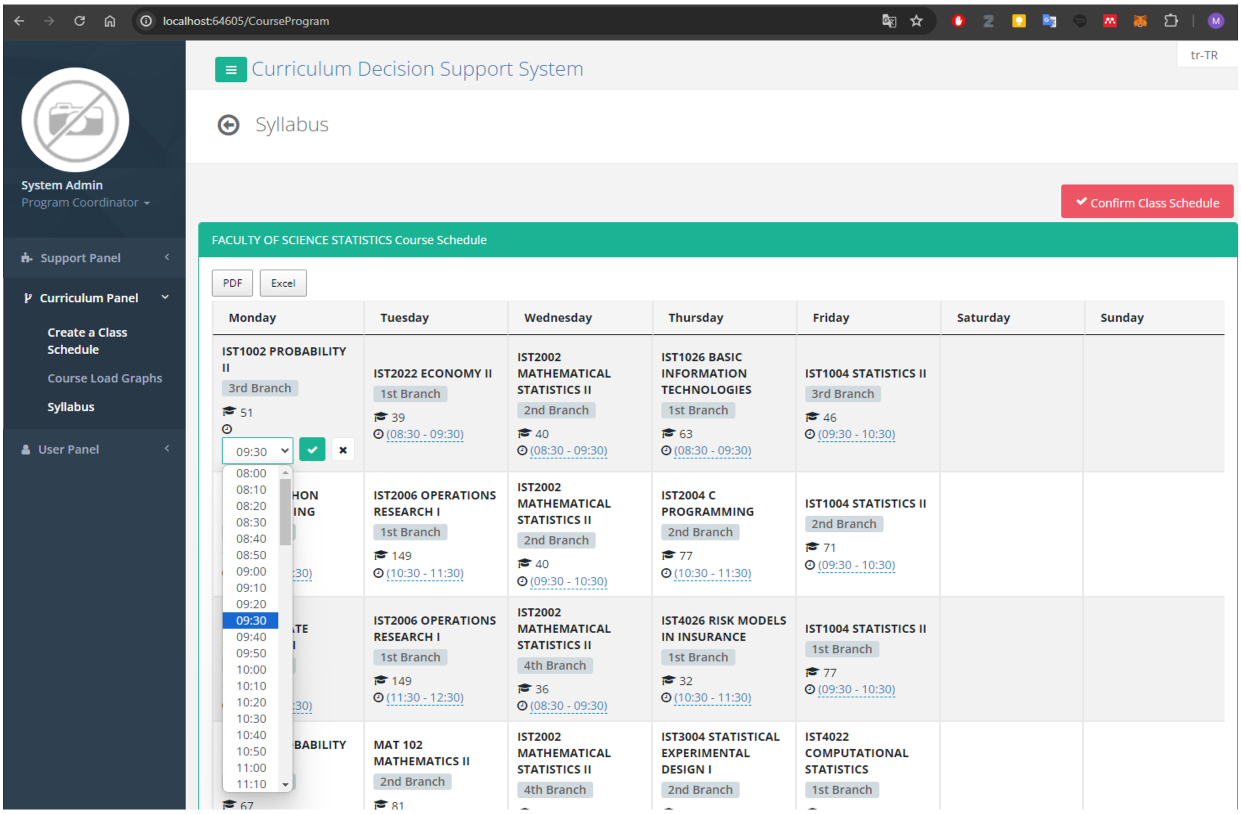

Figure 7. Timetable edit screen

4. CONCLUSION

In this paper, we presented the development and implementation of a web-based Decision Support System (DSS) designed to optimize course timetabling in an online education environment using a simulated annealing algorithm. The sudden shift to online learning due to the COVID-19 pandemic introduced significant challenges, particularly in managing bandwidth and ensuring the efficient operation of online education systems. Our DSS addresses these challenges by balancing the distribution of courses throughout the week, effectively minimizing peak server loads and enhancing the reliability of online course delivery.

The experimental results demonstrate that the DSS significantly improves the distribution of concurrent connections compared to manually generated timetables. By reducing the maximum number of simultaneous connections by approximately 66% and lowering the standard deviation of connections across time slots by around 95%, the system effectively mitigates bandwidth issues and prevents server overloads. This balanced distribution enhances the online learning experience for both students and instructors by reducing technical disruptions and ensuring consistent access to course materials.

The integration of the DSS with the university's existing Student Information System (SIS) and Learning Management System (LMS) facilitates seamless data synchronization and real-time updates, streamlining the timetabling process. The user-friendly interface allows program coordinators to adjust schedules according to specific departmental needs while adhering to overall optimization goals. The flexibility of the system ensures that institutional policies and individual preferences can be accommodated without compromising the efficiency of the timetable.

While the DSS has proven effective in addressing the immediate challenges posed by the transition to online education, there are opportunities for further enhancement. Future work could explore the incorporation of additional constraints and preferences, such as accommodating time zone differences for international students or integrating adaptive learning schedules based on student performance data. Additionally, expanding the algorithm to incorporate machine learning techniques could further optimize the timetabling process by predicting peak usage times and adjusting schedules proactively.

In conclusion, the proposed DSS offers a scalable and effective solution for managing course timetabling in online education environments. By leveraging the simulated annealing algorithm within a flexible and integrated system, we have addressed critical challenges in bandwidth management and schedule optimization. This work contributes to the broader field of educational technology by providing a practical tool that enhances the quality and reliability of online education, ultimately supporting institutions in delivering effective learning experiences in the digital era.

ACKNOWLEDGEMENT

This study was presented to the audience as a model proposal with a local search algorithm at the '8th International Congress on Engineering and Technology Management'.

DECLARATION OF ETHICAL STANDARDS

The authors of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Mevlüt Uysal: Development of the DSS, Conceptualization, Writing - Original Draft

Onur Ceran: Writing - Review & Editing

Mustafa Tanrıverdi: Writing - Review & Editing

Erdal Özdoğan: Writing - Review & Editing

Mutlu Tahsin Üstündağ: Writing - Review & Editing

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] A. Aristovnik, K. Karampelas, L. Umek, and D. Ravšelj, "Impact of the COVID-19 pandemic on online learning in higher education: a bibliometric analysis," ***Frontiers in Education***, 8, 1225834, (2023).

[2] E. Geçer and H. Bağci, "Examining students' attitudes towards online education during COVID-19: evidence from Turkey (Análisis de las actitudes de los estudiantes hacia la educación en línea durante la pandemia de COVID-19. Evidencia de un estudio realizado en Turquía)," ***Cultura y Educacion***, 34(2), 297–324, (2022).

[3] V. R. Ivanova, "Online Training in Higher Education: an Alternative during COVID-19. Strengths and Weaknesses of Online Training," ***Strategies for Policy in Science and Education-Strategii na Obrazovatelnata i Nauchnata Politika***, 29(3), 263–275, (2021).

[4] X. Wang, W. Chen, H. Qiu, A. Eldurssi, F. Xie, and J. Shen, "A Survey on the E-learning platforms used during COVID-19," in 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020, Institute of Electrical and Electronics Engineers Inc., 808–814, (2020).

[5] M. G. Güler and E. Geçici, "A decision support system for scheduling the shifts of physicians during COVID-19 pandemic," ***Computers and Industrial Engineering***, 150, (2020).

[6] F. Biwer et al., "Changes and Adaptations: How University Students Self-Regulate Their Online Learning During the COVID-19 Pandemic," ***Frontiers in Psychology***, 12, (2021).

[7] T. Favale, F. Soro, M. Trevisan, I. Drago, and M. Mellia, "Campus traffic and e-Learning during COVID-19 pandemic," ***Computer Networks***, 176, (2020).

[8] J. Cullinan, D. Flannery, J. Harold, S. Lyons, and D. Palcic, "The disconnected: COVID-19 and disparities in access to quality broadband for higher education students," ***International Journal of Educational Technology in Higher Education***, 18(1), (2021).

[9] R. Bansal, A. Gupta, R. Singh, and V. K. Nassa, "Role and impact of digital technologies in E-learning amidst COVID-19 pandemic," in Proceedings - 2021 4th International Conference on Computational Intelligence and Communication Technologies, CCICT 2021, Institute of Electrical and Electronics Engineers Inc.,194–202, (2021).

[10] G. Korkmaz and Ç. Toraman, "Are We Ready for the Post-COVID-19 Educational Practice? An Investigation into What Educators Think as to Online Learning," ***International Journal of Technology in Education and Science***, 4(4), 293–309, (2020).

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

[12] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," ***Applied Intelligence***, 51(1), 467–492, (2020).

[13] H. Altunay and T. Eren, "A literature review for course scheduling problem," ***Pamukkale University Journal of Engineering Sciences***, 23(1), 55–70, (2017).

[14] M. Hosny, "Metaheuristic Approaches for Solving University Timetabling Problems: A Review and Case Studies from Middle Eastern Universities," ***Smart Innovation, Systems and Technologies***, 111, 10–20, (2019).

[15] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," ***European Journal of Operational Research***, 176(1), 177–192, (2007).

[16] M. Chen, X. Tang, T. Song, C. Wu, S. Liu, and X. Peng, "A Tabu search algorithm with controlled randomization for constructing feasible university course timetables," ***Computers and Operations Research***, 123, (2020).

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

[18] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf, and T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," ***Computers and Operations Research***, 65, 83–92, (2016).

[19] H. Erdoğan Akbulut, F. Ozçelik, and T. Saraç, "A simulated annealing algorithm for the faculty-level university course timetabling problem," ***Pamukkale University Journal of Engineering Sciences***, 30(1), 17–34, (2024).

[20] K. Xiang, X. Hu, M. Yu, and X. Wang, "Exact and heuristic methods for a university course scheduling problem," ***Expert Systems with Applications***, 248, 123383, (2024).

[21] D. Romaguera, J. Plender-Nabas, J. Matias, and L. Austero, "Development of a Web-based Course Timetabling System based on an Enhanced Genetic Algorithm," ***Procedia Computer Science***, 234, 1714–1721, (2024).

[22] A. Kiefer, R. F. Hartl, and A. Schnell, "Adaptive large neighborhood search for the curriculum-based course timetabling problem," ***Annals of Operations Research***, 252(2), 255–282, (2017).

[23] E. Rappos, E. Thiémard, S. Robert, and J. F. Hêche, "A mixed-integer programming approach for solving university course timetabling problems," ***Journal of Scheduling***, 25(4), 391–404, (2022).

[24] M. Mokhtari, M. Vaziri Sarashk, M. Asadpour, N. Saeidi, and O. Boyer, "Developing a Model for the University Course Timetabling Problem: A Case Study," ***Complexity***, 2021, (2021).

[25] G. Colajanni and P. Daniele, "A new model for curriculum-based university course timetabling," ***Optimization Letters***, 15(5), 1601–1616, (2021).

[26] S. Daskalaki and T. Birbas, "Efficient solutions for a university timetabling problem through integer programming," ***European Journal of Operational Research***, 160(1), 106–120, (2005).

[27] M. Lindahl, A. J. Mason, T. Stidsen, and M. Sørensen, "A strategic view of University timetabling," ***European Journal of Operational Research***, 266(1), 35–45, (2018).

[28] N. C. F. Bagger, G. Desaulniers, and J. Desrosiers, "Daily course pattern formulation and valid inequalities for the curriculum-based course timetabling problem," ***Journal of Scheduling***, 22(2), 155–172, (2019).

[29] L. R. Foulds and D. G. Johnson, "SlotManager: A microcomputer-based decision support system for university timetabling," ***Decision Support Systems***, 27(4), 367–381, (2000).

[30] J. Miranda, P. A. Rey, and J. M. Robles, "udpSkeduler: A Web architecture based decision support system for course and classroom scheduling," ***Decision Support Systems***, 52(2), 505–513, (2012).

[31] A. W. Siddiqui, S. A. Raza, and Z. M. Tariq, "A web-based group decision support system for academic term preparation," ***Decision Support Systems***, 114, 1–17, (2018).

[32] T. İnan and A. Fevzi BABA, "Ticari Gemiler İçin Seyir Süresi ve Yakıt Tüketiminin Azaltılması Amaçlı, Hava ve Deniz Şartlarına Göre Rota Optimizasyonu Sistemi (Ege Denizi Örneği)," ***Politeknik Dergisi***, 24(3), 879–892, (2021).

[33] E. Şener, A. S. Sağlam, and F. Çavdur, "Otonom-Paylaşımlı Araç Yönetim Sistemi," ***Politeknik Dergisi***, 26(1), 81–92, (2023).

[34] Ç. Kılıkçıer and E. Yılmaz, "Trafik Işığı Tespiti Yapan Bir Sürücü Güvenlik Destek Sistemi," ***Politeknik Dergisi***, 21(2), 419–426, (2018).

[35] C. Barnhart, D. Bertsimas, A. Delarue, and J. Yan, "Course Scheduling Under Sudden Scarcity: Applications to Pandemic Planning," ***Manufacturing and Service Operations Management***, 24(2), 727–745, (2021).

[36] A. B. Şimşek, "A course timetabling formulation under circumstances of online education," ***Journal of Turkish Operations Management***, 2(5), 781–791, (2021).

[37] C. Cardonha, D. Bergman, and R. Day, "Maximizing student opportunities for in-person classes under pandemic capacity reductions," ***Decision Support Systems***, 154, 113697, (2022).

[38] N. M. Arratia-Martinez, C. Maya-Padron, and P. A. Avila-Torres, "University Course Timetabling Problem with Professor Assignment," ***Mathematical Problems in Engineering***, 2021(1), 6617177, (2021).

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," ***Science***, 220(4598), 671–680, (1983).