Available online at www.garmian.edu.krd

Passer Journal of Basic and Applied Sciences

Passer 8 (Issue 2) (2026) 866-877

*Original Article*

# Adaptive Web Course Scheduling via Multi-Armed Bandit Selection of ACO and PSO Algorithms

**Bonda Sisephaputra*[1], Saifudin Yahya[1], Durrotun Nashihin[1], Azis Suroni[1], Khoirul Islam[1], Harun Al Rasyid[1]**

[1]*Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia.*

Received 22 January 2026; revised 10 April 2026;
Accepted 28 April 2026; available online 29 July 2026

DOI: 10.24271/PSR.2026.572059.2546

**ABSTRACT**

The scheduling of courses in higher education is a complicated combinatorial optimization issue that has to meet many institutional constraints and balance the preferences of lecturers and the availability of resources. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are metaheuristic algorithms that have been extensively used to solve this problem, but their performance frequently changes in response to the nature of the scheduling situation. This paper proposes a dynamic scheduling model based on a Multi-Armed Bandit (MAB) mechanism that is used to dynamically choose between ACO and PSO in the optimization process. The suggested solution will enhance the strength of the algorithms by integrating the strengths of the two algorithms without compromising on the computational efficiency. The framework is applied in a web-based scheduling system which facilitates interactive data management, constraint setup, and lecturer-specific scheduling requests. Real data of faculty scheduling involving sixteen study programs, almost a hundred courses per program, and forty-one lecturers were used as experimental data. The evaluation involves quantitative experiments and qualitative validation of the lecturer surveys. Findings indicate that the adaptive framework can provide a balanced trade-off between the quality of solutions and the performance of the runtime with positive user feedback in the aspects of usability, efficiency, and the acceptance of the system. These results reveal the possibility of the suggested system to be implemented in real-life academic scheduling settings.

https://creativecommons.org/licenses/by-nc/4.0/

*Keywords: Course Scheduling, Metaheuristic Optimization, Multi-Armed Bandit, Ant Colony Optimization, Particle Swarm Optimization.*

## 1 Introduction

Higher education course scheduling is a complicated combinatorial optimization problem, which involves allocating courses, lecturers, classrooms and time slots and meeting a variety of institutional constraints. The issue is generally defined as NP-hard, as the size of the set of possible combinations of schedules grows exponentially with the size of the number of courses, lecturers, and rooms. In real-world academic settings, scheduling systems are required to meet both hard and soft constraints, including avoiding lecturer and room conflicts and lecturer availability, workload balance, and preferred teaching times. These necessities render manual scheduling ineffective and subject to mistakes especially in institutions where there are numerous study programs and numerous courses [1-6].

* Corresponding author
*E-mail address* bondasisephaputra@unesa.ac.id (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

To address these issues, researchers have widely used metaheuristic optimization algorithms to the university course timetabling problem. Compared to more accurate techniques of optimization such as integer programming or constraint programming, metaheuristic algorithms are more scalable to large real-world scheduling problems. Ant Colony Optimization (ACO) is one of those techniques and it has been extensively used due to the pheromone-based search strategy that allows the algorithm to gradually enhance the good scheduling patterns by learning collectively. Previous studies have determined that ACO can be applied to large combinatorial search spaces, and generate high quality schedules with less constraint violations [7-9].

Particle Swarm Optimization (PSO) is another method that has been widely implemented and is based on the social behavior of bird flocks and fish schools. In PSO, the candidate solutions modeled as particles update their positions in a cyclic manner depending on individual experience and global best solutions discovered by the swarm. This process allows PSO to quickly reduce to the promising areas of the search space and is thus especially useful in large-scale optimization problems [10-12].

Although these algorithms are effective, the performance of individual metaheuristic methods is usually sensitive to the nature of the scheduling problem. There are algorithms that can generate quality schedules but take more time to compute and those that converge faster but can give sub-optimal results. This is a trade-off between the quality of solutions and computational efficiency that has prompted researchers to consider hybrid and adaptive optimization methods that integrate the advantages of more than one algorithm. Recent research has explored hybrid metaheuristic models to enhance the robustness and quality of the solutions in scheduling [13-18].

Along with hybrid optimization methods, adaptive algorithm selection strategies have become of interest in the recent past in solving complex optimization problems. The Multi-Armed Bandit (MAB) framework is one of the promising strategies that have their roots in the field of reinforcement learning and are used to model the problem of choosing among a variety of strategies and balancing between exploration and exploitation. Under this system, the system constantly tests the performance of the candidate algorithms and dynamically chooses the one that is likely to give the best result based on past observations. The MAB-based methods have been effectively used in other fields such as recommendation systems, adaptive computing environments, and algorithm configuration [19-21]. Nevertheless, there is a limited use of MAB as a dynamically chosen metaheuristic algorithm in the scheduling of academic courses.

Inspired by these issues, this paper will suggest an adaptive course scheduling framework that will dynamically choose between the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) with the help of a Multi-Armed Bandit mechanism. The suggested framework will attempt to integrate the merits of the two algorithms in that the system will be able to utilize the quality solution generation of ACO and enjoy the quick convergence ability of PSO.

Besides the optimization framework, the given research also applies the proposed approach to a web-based academic scheduling application that can be used in practice by an institution. The system assists in the management of scheduling information, the availability of lecturers, classroom assignment, and user-specified constraints like preferred teaching hours or course specifications. The given framework is tested on the basis of the real scheduling data of a faculty comprising of sixteen study programs, 172 courses, 41 lecturers, and 35 classrooms, which will give a realistic setting to test the performance of the adaptive scheduling approach.

The main contributions of this study can be summarized as follows:

1. The development of an adaptive scheduling framework that dynamically selects between Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) using a Multi-Armed Bandit (MAB) mechanism.
2. The integration of the adaptive optimization framework into a web-based academic scheduling system that supports institutional constraints and lecturer-specific scheduling preferences.
3. An empirical evaluation using real-world faculty scheduling data combined with lecturer feedback, enabling both technical and usability assessment of the proposed scheduling system.

The remainder of this paper is organized as follows. Section 2 presents the proposed methodology, including the adaptive MAB-based algorithm selection mechanism and the system architecture. Section 3 discusses the experimental results and evaluation. Finally, Section 4 concludes the paper and outlines possible directions for future research.

## 2 Related Works

The problem of university course scheduling has received a lot of research in the area of combinatorial optimization because of its complexity and real-life significance in educational establishments. The issue is the allocation of courses, lecturers, classrooms and time slots and meeting a set of hard and soft constraints. Different optimization methods have been suggested to solve this issue in the last decades, such as precise methods, heuristic methods, and metaheuristic algorithms. Metaheuristic optimization techniques have emerged as some of the most popular methods of these techniques because they can search large search spaces and generate high quality feasible schedules in a reasonable amount of computational time [1, 4, 5, 6, 22].

Genetic algorithms (GA) and other evolutionary computation methods were often used in early research on university timetabling. These methods could produce viable schedules through successive enhancement of candidate solutions according to fitness evaluation functions. Indicatively, Mahlous and Mahlous Mahlous [3] suggested a genetic algorithm which takes into consideration the preferences of students in the scheduling process and it was shown that evolutionary algorithms can be effective in balancing various scheduling constraints. Nevertheless, genetic algorithms can be highly sensitive to parameter tuning, and can be slow to converge in large scheduling problems.In order to overcome these shortcomings, scholars have considered swarm intelligence-based optimization algorithms, specifically, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). ACO has been extensively used in the context of timetabling problems due to the pheromone-based search mechanism, which allows the algorithm to acquire the promising scheduling patterns by exploration and reinforcement. It has been demonstrated in the previous literature that ACO can be successfully used to minimize conflict violations without sacrificing solution diversity in large scheduling problems [7-9].

However, ACO can take a long time to compute because it is an iterative process of updating pheromones.

Educational timetabling problems have also been successfully solved using Particle Swarm Optimization (PSO). In PSO, candidate solutions are modeled as particles that move in the search space according to their experience and the experience of the swarm. The mechanism enables PSO to converge quickly to promising solutions, which makes it especially appropriate to large-scale optimization problems. A number of studies have already stated that PSO-based methods are capable of producing workable schedules with relatively quick execution duration than other metaheuristic algorithms [10-12, 23, 24].

More recently, scientists have explored hybrid metaheuristic methods to integrate the capabilities of several optimization algorithms. The hybrid frameworks are designed to enhance the quality of solutions and computational efficiency through the combination of various search strategies. As an illustration, hybrid metaheuristic methods to complex scheduling problems have been suggested in recent research and have shown that a combination of several optimization methods can enhance robustness and scheduling efficiency [13-18, 25]. These methods imply that there is no single best metaheuristic algorithm that can be used in all scheduling situations [16, 17].

Along with the hybrid optimization methods, adaptive algorithm selection mechanisms have become one of the promising directions of research. Adaptive optimization systems seek to dynamically choose the best algorithm according to the nature of the problem or the success of the past optimization steps. These methods are especially applicable to complex combinatorial problems whose performance of the algorithm can be different with various datasets and scheduling requirements.

Multi-Armed Bandit (MAB) model is one such promising adaptive framework that builds on the reinforcement learning and solves the exploration-exploitation trade-off in sequential decision-making problems. Within this framework, a system repeatedly chooses among several candidate strategies and learns based on the reward received after the choice. The MAB-based techniques have been used in various fields such as recommendation systems, algorithm configuration, and adaptive computing environments [19, 26]. These strategies allow systems to dynamically change their optimization strategies depending on observed performance.

Although the adaptive optimization is gaining more and more interest, the use of Multi-Armed Bandit strategies to dynamically choose metaheuristic algorithms in the academic course scheduling is not well studied. The majority of the available literature is dedicated to individual metaheuristic algorithms or pre-defined hybrid systems that do not dynamically choose algorithms. This shortcoming encourages the creation of adaptive scheduling models that can automatically identify the most appropriate optimization algorithm in the process of scheduling.

This paper presents a dynamic course scheduling model, which combines the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) by applying a Multi-Armed Bandit-based algorithm selection mechanism. In contrast to the traditional hybrid methods, which mix algorithms in a predetermined fashion, the proposed method dynamically chooses the most suitable optimization algorithm depending on the observed performance on the scheduling. The goal of this adaptive mechanism is to find a balanced trade-off between the quality of solutions and the efficiency of computations and to be robust in various scheduling conditions.

Recent studies published in the *Passer Journal of Basic and Applied Sciences* have also explored optimization techniques and intelligent systems in solving complex real-world problems. For example, Kakarash et al. [27] applied Ant Colony Optimization in wireless sensor network optimization, demonstrating the effectiveness of swarm intelligence approaches in handling complex optimization scenarios. Similarly, Mansour and Al-hamdani [28] investigated tabu search for multi-objective optimization problems, while other studies explored genetic algorithm-based scheduling [29] and optimization modeling approaches [30] in applied systems. These studies highlight the growing interest in adaptive and metaheuristic optimization techniques within the PASSER research community. However, most existing works focus on single optimization strategies or fixed models. In contrast, the present study introduces a dynamic Multi-Armed Bandit-based framework to adaptively select between multiple metaheuristic algorithms, providing improved flexibility and performance in course scheduling problems.

A comparative summary of typical studies on course timetabling optimization is given in Table 1. The comparison takes into account a number of significant factors, such as the optimization methods used, experimental data, performance measures, and the key limitations of both methods. Based on this comparison, it is possible to notice that the majority of the available studies use single metaheuristic algorithms or fixed hybrid approaches. Nevertheless, dynamically selected adaptive algorithm selection mechanisms which identify the most appropriate optimizer at the time of scheduling are not well studied. The given limitation prompts the creation of the suggested adaptive scheduling framework which is built upon Multi-Armed Bandit guided metaheuristic selection.

**Table 1:** Comparison of Existing Studies on Course Scheduling Optimization.

| Study | Technique | Dataset | Metrics | Limitation |
|---|---|---|---|---|
| Mahlous & Mahlous (2023) | Genetic Algorithm | Student timetabling dataset | Conflict rate, preference satisfaction | Convergence can be slow |

| | | | | |
|---|---|---|---|---|
| Mazlan et al. (2019) | Ant Colony Optimization | University timetable dataset | Constraint violations | High runtime |
| Tassopoulos et al. (2023) | Particle Swarm Optimization | School timetabling dataset | Fitness value, runtime | May converge prematurely |
| Alhussan et al. (2025) | Hybrid Metaheuristic | Scheduling benchmark dataset | Optimization quality | Static algorithm combination |
| Lagos & Pereira (2024) | MAB Hyper-heuristic | Combinatorial optimization dataset | Reward-based performance | Not applied to course scheduling |
| This Study | MAB + ACO + PSO | Real faculty dataset | Conflict rate, runtime, compactness | Adaptive selection |

## 3 Methods and Materials

The research methodology used in this study is a multi-stage research methodology that incorporates an adaptive optimization mechanism, a web-based scheduling system, and an empirical test based on actual academic scheduling data. The aim of the proposed strategy is to produce viable course schedules and meet the constraints of the institution and adjust to the different computational capabilities of the different optimization algorithms.

The data that was used in this research was collected in a faculty with sixteen study programs. The programs usually have two to four parallel classes. All in all, the scheduling data consists of 172 courses, 41 lecturers and 35 classrooms, which should be allocated to the available time slots during an academic week. The process of scheduling has to meet a number of constraints such as the availability of lecturers, the capacity of the classroom, and the evasion of scheduling clashes among classes and groups of students.

This scheduling environment is usually complex, and thus, manual construction of timetables is usually ineffective and subject to errors. Thus, a flexible optimization model is suggested to help administrators to produce realistic and quality schedules. The suggested framework combines a Multi-Armed Bandit (MAB) based algorithm selection mechanism with two metaheuristic optimization algorithms, i.e. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

The general methodology has a number of components: an adaptive algorithm selection mechanism, the system architecture and interaction model, computational complexity analysis, and a real-world scheduling case study to be used in the validation.

### 3.1 Adaptive MAB-Based Method-Selection Mechanism

The initial step of the methodology is the creation of an adaptive selection model based on the multi-armed bandit (MAB) paradigm to select the most efficient scheduling algorithm dynamically during the runtime process as observed in Figure 1. MAB models are especially appropriate to those problems that involve exploration-exploitation trade-offs when the system needs to choose between exploration and exploitation of the best-performing method that has been identified so far [31].

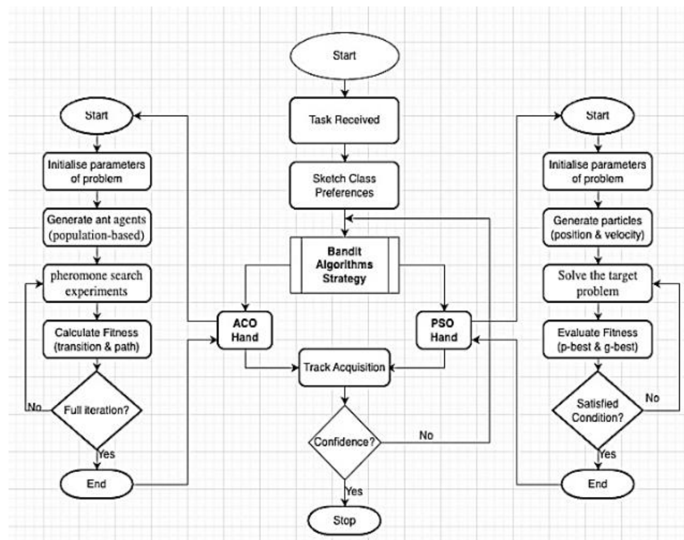


**Figure 1:** MAB-Based Selection Process for ACO and PSO.

ACO and PSO are also candidates in this study and are part of the framework. Nevertheless, the specifics of their mechanisms and mathematical models are not addressed in detail, as the key point is to use the MAB strategy to adaptively decide on the algorithm to be used at various points of the scheduling process. Two popular MAB strategies are used, ε-greedy and Upper Confidence Bound (UCB1) [32-34]. Besides, the reward is based on time performance measures (e.g., conflict reduction, computation time, timetable compactness), so that efficiency, as well as quality, is used to select the best.

#### 3.1.1 ε-Greedy Selection

In the ε-greedy approach, the algorithm selects the current best-performing method with probability $1-\epsilon$, and a random method with probability ε.

$$A_t = \begin{cases} arg\ max_{a \epsilon A}\ \hat{\mu}a\ , with\ prob\ 1-\epsilon \\ rand\ action\ from\ A,\ with\ prob\ \epsilon \end{cases} \quad (1)$$

where:
$A_t$= chosen algorithm at iteration $t$,
$A$ = set of candidate algorithms (e.g., ACO, PSO),
$\hat{\mu}a$ = estimated reward (fitness) of algorithm $a$,
$\epsilon$ = exploration rate

### 3.1.2 Upper Confidence Bound (UCB1)

UCB1 balances exploration and exploitation by selecting the algorithm with the highest upper confidence bound. The second term encourages exploration of less frequently chosen algorithms, ensuring adaptability.

$$A_t = argmax_{a \epsilon A}\left(\widehat{\mu} a + \sqrt{\frac{2\ ln\ t}{N_a}}\right) \qquad (2)$$

where:
$\widehat{\mu} a$ = estimated average reward of algorithm $a$,
$N_a$ = number of times algorithm $a$ has been selected,
$t$ = current iteration.

The suggested system is compared with the fixed-algorithm scheduling methods on the baseline. There are three main metrics that are used. The first one is conflict rate, which is the number of overlapping sessions compared to number of planned sessions. The second one is computation time, which is the amount of time taken to generate a complete timetable. Finally, the last one is compactness score, which is density of the timetable, larger scores mean fewer idle intervals between sessions. The experiment is repeated ten times to ensure that stochastic variations that are common to the metaheuristic methods are taken into consideration. The Wilcoxon signed-rank test is used to establish the statistical significance of the performance differences.

## 3.2 System Interaction and Architecture

In order to illustrate the interaction between the system components of the proposed scheduling framework more clearly, the operational workflow may be represented in terms of a sequence-based interaction model. The sequence diagram explains the interaction between the users and the web-based scheduling application and the system dynamically choosing the optimization algorithm using the Multi-Armed Bandit (MAB) mechanism.

First, the administrator offers scheduling information such as course information, lecturer assignments, classroom availability, and scheduling constraints via the web interface. The input information is then recorded in the database by the system and the scheduling environment is ready. After the scheduling process is initiated, the optimization engine asks the MAB controller to decide which algorithm is to be applied in the current optimization iteration.

Depending on the exploration-exploitation strategy chosen, the MAB controller will assess the historical performance of the candidate algorithms and choose between Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The chosen algorithm then produces a candidate timetable solution that is measured based on specific performance measures, including conflict reduction, runtime, and compactness of the schedule.

The performance score produced after assessment is sent back to the MAB controller as a reward signal. This reward is employed to revise the approximated performance of the chosen algorithm, so that the system can enhance the subsequent algorithm selection choices. Lastly, the optimal schedule created in the optimization process is saved in the database and made available to the users on the web interface.

The system interaction in the scheduling process is shown in Figure 2. This is initiated by the administrator entering the scheduling information using the web interface with the course details, lecturer assignments and classroom availability. The scheduling engine then asks the Multi-Armed Bandit (MAB) controller to decide what optimization algorithm is to be run. Depending on the outcome of the selection, the Ant Colony Optimization (ACO) algorithm or Particle Swarm Optimization (PSO) algorithm is implemented to produce a candidate timetable. The resultant schedule is assessed and stored in the database and made available to the user via the system interface.

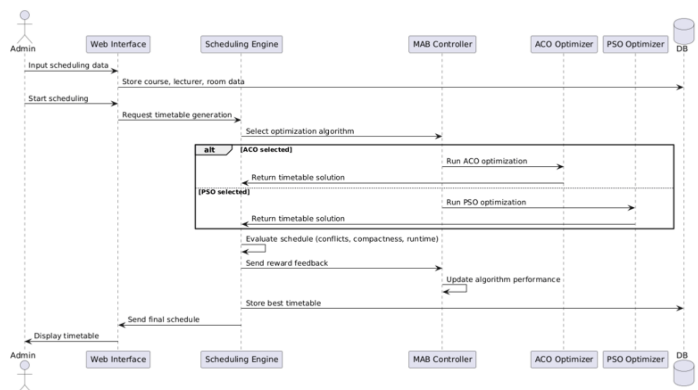


**Figure 2:** Sequence diagram illustrating the interaction between system components during timetable generation.

Figure 3 demonstrates the general workflow of the suggested adaptive course scheduling framework. It starts by receiving and pre-processing scheduling information, such as courses, lecturers, classrooms, and time slots available. Multi-Armed Bandit (MAB) controller then decides dynamically the optimization algorithm to be implemented, either Ant Colony Optimization (ACO) or Particle Swarm Optimization (PSO). The chosen algorithm produces a candidate schedule, which is rated against predetermined performance indicators like conflict rate and compactness of the timetable. The outcome of the evaluation is taken as a feedback to revise the MAB model, and the system is able to enhance the selection of algorithms in future and achieve the optimal timetable solution.

The proposed web-based scheduling framework has a system architecture depicted in Figure 4. The system comprises of a data management user interface, a scheduling engine that makes timetables, and a Multi-Armed Bandit (MAB) controller that is able to choose the most appropriate optimization algorithm dynamically. The MAB controller decides which one among Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) is to be implemented based on their past performance. The algorithm chosen produces candidate schedules which are tested based on predefined performance metrics. The most optimal schedule is then saved in the database and is displayed to the users on the web interface.

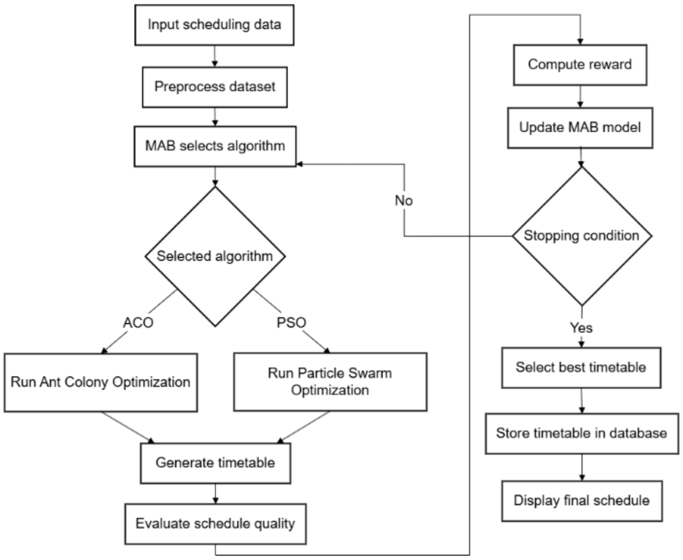


**Figure 3:** Workflow of the proposed adaptive course scheduling framework.

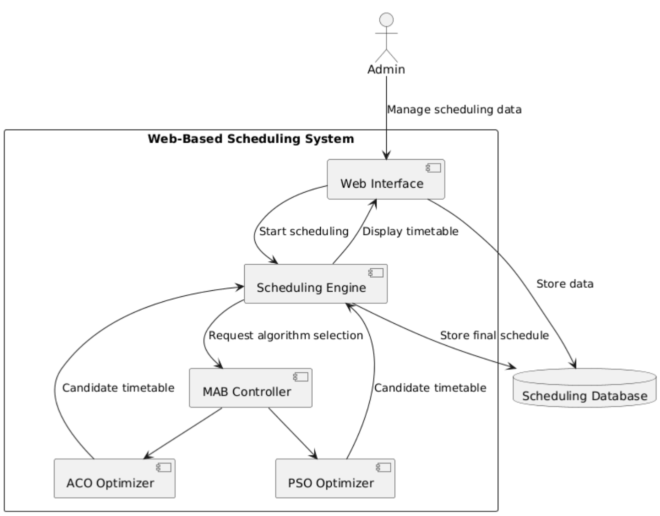


**Figure 4:** Architecture of the web-based adaptive course scheduling system integrating the MAB controller with ACO and PSO optimizers.

## 3.3 Integration into Web-Based Scheduling Application

The second stage will be the application of the adaptive MAB mechanism to a web-based course scheduling system to make the practical application possible in the academic environment. It is implemented on the Django web application and Flutter web application that offers scalability and maintainability. The backend does the computation of MAB, the implementation of scheduling algorithms and special user-defined constraints, e.g. lecturer-preferred time, course sequencing or class-grouping requests. The frontend provides the administrators and lecturers with an interactive interface to specify institutional regulations and personal limitations, see the created schedules, and get the outcomes. One layer that links the MAB decision-making engine and the scheduling modules is the RESTful API layer that enables modular updates without interfering with the system architecture. Not only does this combination conform to best practice in educational scheduling software design, it also introduces flexibility to user requested constraints, which increases its usability and responsiveness in real-time [35, 36].

## 3.4 Computational Complexity Analysis

The optimization algorithms that are applied in the timetable generation process are the main determinants of the computational overhead of the proposed scheduling framework. The scheduling engine in this research employs two metaheuristic algorithms, that is, an Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) which search the search space to find feasible timetable solutions.

Overall, ACO is more computationally intensive, as several ants are used to build candidate solutions and modify the values of pheromones at each iteration. On the contrary, PSO is usually quicker because the calculation of the position and velocity of the particles is rather simple. Consequently, PSO tends to converge faster whereas ACO tends to give better solutions in complicated scheduling problems.

Multi-Armed Bandit (MAB) mechanism has only a small incremental computational cost since it merely chooses the algorithm according to the past performance. The selection process is cheap in terms of computation as compared to the optimization process itself due to the limited number of candidate algorithms in this study.

In general, the optimization algorithms applied to timetable generation dominate the computational cost of the proposed framework, and the adaptive MAB selection mechanism is associated with a small overhead. This design allows the system to strike the balance between the quality of the solutions and the computational efficiency when scheduling.

## 3.5 Real-World Scheduling Case Study

To show the relevance of the proposed framework in a real academic setting, the scheduling system was tested on a dataset that was received in a faculty that included sixteen study programs. The data has 172 courses, 41 lecturers and 35 classrooms that need to be distributed among various time slots in the academic week.

The time schedule is arranged into 18 time slots in a day, morning to evening sessions. Every course should be allocated to a suitable lecturer, classroom and time slot without violating institutional constraints like availability of lecturers, classroom capacity and avoiding time clashes.

As an example, in the process of scheduling, the system can allocate a course like Data Structures to a given lecturer and classroom at a given time slot. The optimization algorithms search through the different combinations of the assignments and compare them using the performance metrics that are set like conflict rate and compactness of the timetable.

The search space is very large because of the high number of possible combinations of schedules, and therefore manual scheduling is difficult and time-consuming. The adaptive MAB-based optimization model solves this problem by dynamically choosing the most appropriate optimization algorithm in the scheduling process to effectively produce high-quality timetables.

### 3.6 Survey and Feedback Analysis

The evaluation phase is intended to gather empirical data through a survey methodology rather than the system implementation. The questionnaire was distributed via Google Forms and sent to lecturers, students, and academic staff involved in course scheduling. The questionnaire was designed to measure:

1. Organizational (e.g., the availability of lecturers, the capacity of rooms, scheduling),
2. User preferences (e.g., schedule compactness, fairness, flexibility), and
3. Adaptive optimizer perceptions (e.g., expected benefits, usability, and feasibility).

The answers were measured using a 5-point Likert scale, and it became possible to measure them quantitatively and get qualitative feedback. The received data will provide the information on the current scheduling problems and expectations of the stakeholders that will serve as a valuable validation of the proposed adaptive MAB-based framework.

## 4 Results and Discussions

This part reports the findings of the research according to its key contributions, including the performance of the adaptive method of choosing the MAB, integrating the hybrid optimizer into a web-based academic scheduling tool with the support of special constraints, and the survey-based analysis that reflects the stakeholder attitudes on academic scheduling.

### 4.1 Performance of Adaptive MAB-Based Metaheuristic Selection

Table 2 presents a summary of the conflict rates (penalty scores) of ACO, PSO, and adaptive MAB-ACO+PSO approach. As can be seen, ACO always finds better-quality schedules, and the average penalty is about 5 and the optimal maximum is 2, which is indicative of its pheromone-informed solution construction which prefers viable schedules with minimal conflicts. By comparison, PSO is not doing well at conflict reduction, with an average of about 9 and a best case of 8, which is consistent with previous results that PSO can quickly converge to suboptimal solutions in combinatorial problems. The adaptive MAB-based approach demonstrates both a superior and a balanced performance with an average of approximately 4 and a highest score of 3, which shows the superiority of the dynamic algorithm choice in exploiting the complementary benefits of ACO and PSO.

**Table 2:** Conflict Rate Comparison of Scheduling Methods.

| Method | Average | Best |
|---|---|---|
| ACO | ~5 | 2 |
| PSO | ~9 | 8 |
| MAB-ACO+PSO | ~4 | 3 |

Table 3 shows performance at runtime. PSO is the quickest, and it only takes 18-22 seconds, which is in line with its quick convergence features. ACO is computationally more expensive with a run time of approximately 48-68 seconds because it uses iterative pheromone updates. Surprisingly, the MAB-ACO+PSO hybrid attains intermediate runtimes (33-36 seconds), indicating that adaptivity is a viable concept that can be used to improve robustness.

**Table 3:** Runtime Comparison of Scheduling Methods.

| Method | Average | Best |
|---|---|---|
| ACO | ~68s | 48s |
| PSO | ~22s | 18s |
| MAB-ACO+PSO | ~36s | 33s |

In Table 4, strengths and weaknesses are concisely summarized:

- ACO: Excellent solution quality, but slow.
- PSO: Fast, yet lower quality.
- MAB-ACO+PSO: Stable and robust, effectively balancing both aspects.

These results support the hypothesis that the adaptive MAB-based approach, by dynamically balancing exploration (PSO's speed) and exploitation (ACO's quality), yields consistently strong performance across varying scheduling scenarios, an essential trait for real-world academic timetabling.

**Table 4**: Performance.

| Method | Strengths | Weaknesses |
|---|---|---|
| ACO | Best in quality | Big runtime |
| PSO | Converges fast | Less quality |
| MAB-ACO+PSO | Stable and robust | Balance |

### 4.2 Runtime Evaluation under Different Scheduling Scenarios

To further test the performance of the proposed adaptive scheduling framework, other experiments were performed to test the execution time of the optimization algorithms in various scheduling conditions. As the problem of scheduling courses may be very large and complex in different cases, based on the number of courses, lecturers, and classrooms, it is necessary to consider the scalability of the suggested solution.

Three scheduling situations were taken into account in this study depending on various dataset sizes. The former is a minor scheduling issue with few courses and classrooms. The second case is a medium-sized data set that is related to a normal departmental scheduling problem. The third case is the complete faculty data that was employed in this research consisting of 172 courses, 41 lecturers, and 35 classrooms. Table 5 shows the mean time taken by ACO, PSO, and adaptive MAB-based method to produce feasible schedules in each of the above cases.

**Table 5:** Average Runtime Required by ACO, PSO, and the adaptive MAB-based.

| Scenario | Dataset Size | ACO Runtime | PSO Runtime | MAB Adaptive Runtime |
|---|---|---|---|---|
| Small | ~50 courses | 18 s | 9 s | 12 s |
| Medium | ~100 courses | 41 s | 17 s | 25 s |
| Large | 172 courses | 68 s | 22 s | 36 s |

The findings indicate that PSO is always the fastest to run because it has a quick convergence mechanism. Nevertheless, ACO is more likely to take more time to execute due to the iterative nature of its pheromone update mechanism. Adaptive MAB-based approach is the intermediate in terms of runtime performance as it dynamically chooses the best algorithm in the process of scheduling. This action allows the proposed framework to balance computational efficiency and quality of solutions.

## 4.3 Statistical Significance Analysis

Statistical significance analysis was conducted using the Wilcoxon signed-rank test. Since metaheuristic algorithms involve stochastic processes, multiple runs were performed under identical conditions. The Wilcoxon test was applied to evaluate whether the differences in performance between algorithms were statistically significant rather than due to random variation.

The statistical test results suggest that the adaptive MAB-based method has statistically significant better results than the standalone PSO algorithm in the conflict reduction. The comparison of adaptive approach and ACO presents similar performance in terms of quality of schedule and lower runtime in various experimental runs. These findings imply that adaptive scheduling framework is a good combination of ACO and PSO. The system can enhance robustness and achieve consistent scheduling performance across runs by dynamically choosing the best algorithm to use in the optimization process.

## 4.4 Integration and Usability of the Web-Based Scheduling System

The second contribution lies in the deployment of the scheduling optimizer into a web-based platform, providing practical usability for academic administrators. The system supports interactive data entry, constraint management, and special requests, making it adaptable to institutional needs.

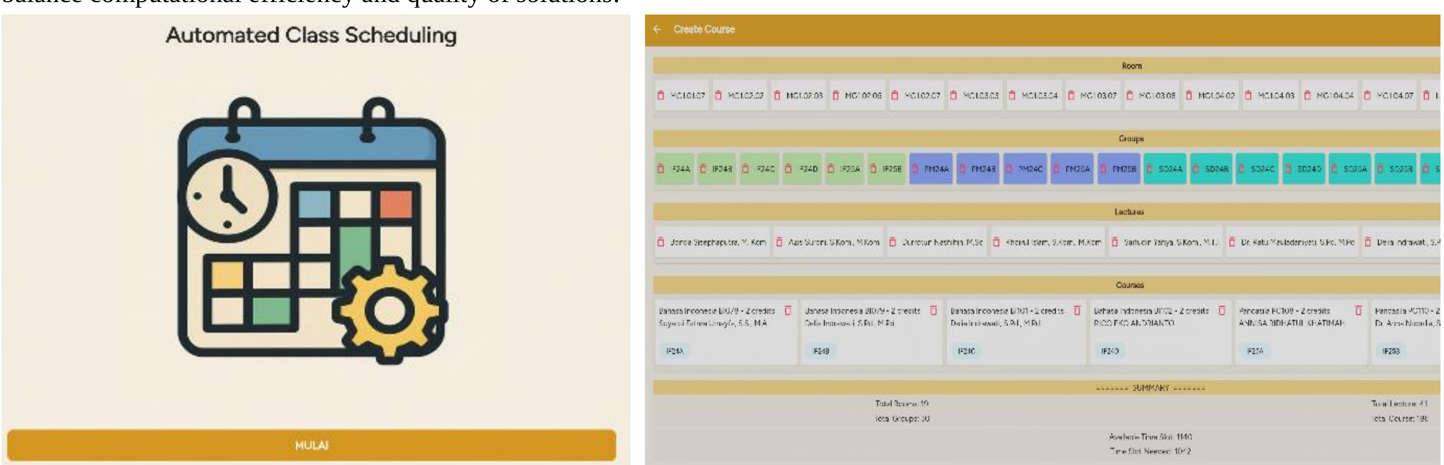


**Figure 5:** Intro page of the scheduling system (left) and Input page for rooms, groups, and lecturers (right).

Figure 5 (left) shows the System Entry feature. This page illustrates the introductory page providing a simple starting interface for users, ensuring accessibility for both administrative staff and lecturers. Next, the data input can be shown in Figure 5 (right). This feature is called Data Input feature, where administrators can input rooms, groups, lecturers, and courses, defining the scheduling dataset. This modular approach enables easy updates and scalability across multiple programs. After that, figure 6 (left) depicts the Summary Data feature. Once data are entered, the system automatically generates a summary of available resources (rooms, groups, lecturers, total courses). This ensures users are aware of scheduling constraints before optimization. Special Request Handling then is shown in Figure 6 (right).

This figure is a special feature of the system that enables users to make special requests, e.g. the time a lecturer prefers to work or that a course is mandatory. This aspect improves fairness and flexibility over the conventional automated scheduling tools.

Figure 7 represents the Final Timetable. Once the optimization has been completed, the timetable generated is provided in a weekly view organized into rooms and period, showing the course allocations. There is a reduction in conflicts and special requests are followed, which proves the efficiency of the integrated optimizer. In general, the integration does not only prove the feasibility of the adaptive MAB mechanism, but also demonstrates the importance of user-centered design in scheduling systems. Similar studies highlight the importance of dealing with user-defined constraints when implementing scheduling systems in real institutional settings [37, 25].

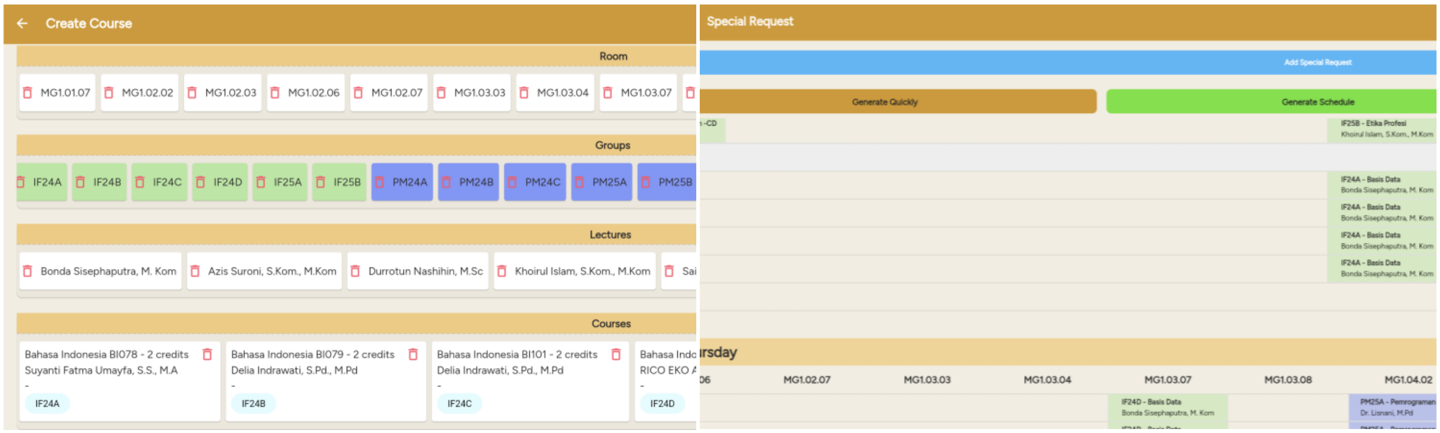


**Figure 6:** Course creation and dataset summary (left), special request interface (right).


**Figure 7:** Final optimized timetable.

## 4.5 Analysis and Implications of Evaluation Results

The satisfaction survey was carried out to assess the user acceptance and perception of the designed scheduling system by using all the 41 lecturers in the faculty under investigation. The survey was evaluated on five dimensions, namely, usability, efficiency, effectiveness, reliability, and overall satisfaction, which were measured with a 5-point Likert scale. Table 6 provides the summary of the results, whereas Figure 8 illustrates the average scores of the dimensions. The results, in general, point to a high level of satisfaction in all aspects, which means that the system does not only provide technically optimal schedules, but it is also well-received by its end users.

**Table 6**:Results of Lecturer Satisfaction Survey.

| Item Code | Statement | Mean |
|---|---|---|
| Usability | | |
| U1 | The system interface is easy to navigate. | 4.35 |
| U2 | The menu and features are logically structured. | 4.25 |
| U3 | The timetable visualization (grid/calendar) is clear. | 4.40 |
| Efficiency | | |
| E1 | The system reduces time required to finalize schedules | 4.15 |
| E2 | The system provides scheduling results faster than manual. | 4.25 |
| E3 | Automatic conflict detection is efficient | 4.50 |
| Effectiveness | | |
| F1 | Adaptive algorithm improves schedule quality | 4.10 |
| F2 | The system generates schedules that are conflict-free | 4.15 |
| F3 | The system balances teaching load fairly among lecturers | 4.0 |
| Reliability | | |
| R1 | The timetable data generated is accurate and reliable | 4.10 |
| R2 | The system is stable and rarely encounters errors | 4.25 |
| R3 | The help/support features are useful. | 4.05 |
| Satisfaction | | |
| S1 | I am satisfied with the overall performance | 4.25 |

The survey results of all 41 lecturers are a clear indication of the practical value of the system. Figure 8 below depicts that the radar chart indicates high ratings in all five dimensions of evaluation: usability, efficiency, effectiveness, reliability, and overall satisfaction with all the mean scores greater than 4.0 on a 5-point Likert scale. This shows that the system is considered valuable and reliable by the stakeholders in the course of scheduling courses. Table 6 shows that usability was rated the highest, with the clearness of timetable visualization (U3 = 4.40) and the ease of navigation (U1 = 4.35) being rated the highest. This confirms the usefulness of user-centered design process to surmount one of the most significant obstacles to adoption in academic scheduling systems, which is overly complex or unintuitive interfaces.

The efficiency was also rated high with the most helpful feature being the automatic conflict detection (E3 = 4.50). The cause of such results could be explained by the purpose of the system, which was to reduce the administrative burden and reduce the time of scheduling. The findings show that the adaptive metaheuristic engine, as well as the real-time conflict management, may offer realistic efficiency benefits in comparison with the manual methods of scheduling. The effectiveness was rated highly (4.0-4.15) and the respondents acknowledged that the adaptive algorithm improves the quality of schedules (F1 = 4.10). However, the equality of the teaching loads is not as high, and it implies that some further enhancement might be demanded to achieve more equal distributions.

The reliability is also performing adequately particularly in the stability of the system (R2 = 4.25). It demonstrates that the implementation is technically valid. Subsequently, the support features (R3 = 4.05) score is less, and it can be enhanced by offering a better quality of user assistance and documentation to enhance trust and adoption. Lastly, the system acceptance is confirmed by the overall scores of the satisfaction being high with the intentions to use the system in the future (S3 = 4.50) and with the perception of the positive impact of the system on the quality of teaching (S2 = 4.25). Overall, these results indicate that the system is prepared to be deployed at the institutional level in the long term and establish the areas, including the workload equity and user friendliness, that can be enhanced.

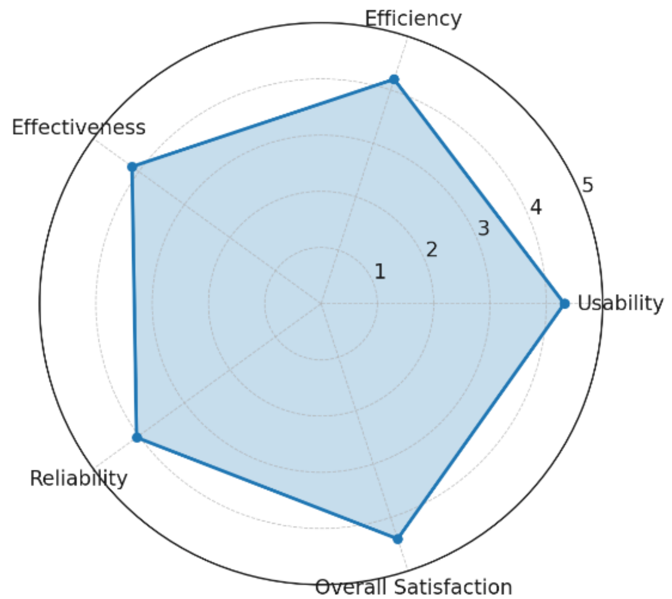


**Figure 8:** Radar chart user satisfaction.

In the future, it is possible to make a number of enhancements that would make the system more effective. To start with, the adaptive algorithm can be optimized to respond more to teaching load fairness so that the courses are distributed more fairly among lecturers. Second, usability might be enhanced by increasing the number of help and support options, including step by step instructions, tutorials, or built-in frequently asked questions, to help new users feel more confident about using the system. Lastly, it would be more scalable and robust to expand the framework to include other institutional constraints (e.g. cross faculty scheduling or changes during mid-semester). Such improvements would not only increase user satisfaction, but also strengthen the practical applicability of the adaptive MAB ACO+PSO scheduling system to larger educational settings.

## Conclusion

In this paper, an adaptive method of scheduling university courses was proposed, which dynamically chooses between the Ant Colony Optimization (ACO) and the Particle Swarm Optimization (PSO) with the help of a Multi-Armed Bandit (MAB) mechanism. The goal of the suggested framework is to enhance the robustness of scheduling by balancing the quality of the solution and the efficiency of the computations when generating the timetable.

The experimental findings indicate that the adaptive MAB-based methodology has a stronger trade-off between the run-time performance and the quality of scheduling than the single-algorithm approaches. Although ACO is more likely to generate quality schedules with minimal conflicts, it takes more time to execute. PSO on the other hand offers quicker convergence but can give suboptimal solutions in certain situations. The proposed framework can combine the benefits of both algorithms and produce stable scheduling outcomes by dynamically choosing the most appropriate algorithm according to the observed performance.

The proposed optimization model was also successfully integrated into a web-based scheduling system that supports the institutional constraints and the scheduling preferences that are specified by the users. The system will allow the administrators to manage the scheduling data, generate the best schedules and incorporate special requests such as the availability of lecturers and classroom requirements. The comparison in terms of the actual data on the faculty scheduling and feedback on the users demonstrates that the system has certain realistic benefits in the context of the usability, efficiency, and reliability.

Although these are encouraging outcomes, there are a number of limitations. The existing model is aimed at minimizing conflicts and compactness in terms of schedules as the other aspects of the workload fairness among lecturers may be enhanced. Moreover, the analysis has been based on the data of one faculty setting, which can be considered a limitation to the generalizability of the findings.

The proposed framework can be expanded in future studies with more scheduling constraints, better workload balancing mechanisms, and other adaptive optimization strategies. Moreover, using the suggested method with bigger institutional data sets or a multi-faculty scheduling setting might give more information about the scalability and strength of the adaptive scheduling framework.

## Funding

This research was supported by non-APBN funding from Universitas Negeri Surabaya (UNESA). The authors sincerely extend their gratitude to UNESA for the support, encouragement, and trust that made this research possible. The financial assistance, academic resources, and research facilities that are provided by UNESA have been instrumental in ensuring the successful completion of this work. The authors also acknowledge the conducive academic environment and collaborative spirit fostered by the institution, which greatly contributed to the development and refinement of this study. Without such comprehensive support, the implementation and finalization of this study would not have been possible.

## Authors contribution

**Table 7:** Author Contribution.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bonda Sisephaputra | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | | ✓ | |
| Saifudin Yahya | | ✓ | | | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Durrotun Nashihin | | ✓ | ✓ | ✓ | | | ✓ | | | ✓ | ✓ | | ✓ | ✓ |
| Azis Suroni | | | | ✓ | ✓ | | | ✓ | ✓ | ✓ | | | | ✓ |
| Khoirul Islam | | | | | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ |
| Harun Al Rasyid | | | | | | ✓ | | | | ✓ | ✓ | | | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## Conflict of interests

The authors declare no conflict of interest.

## References

[1] Abdipoor, S., Yaakob, R., Leng, S. & Abdullah, S. Meta-heuristic approaches for the university course timetabling problem. *Intell. Syst. Appl.* **19**, 200253 (2023).

[2] Herath, A. & Alabama, N. A comparative study on solving university timetabling problems with emphasis on genetic algorithms. *Int. J. Sched. Optim.* **25**, 392–408 (2024).

[3] Mahlous, A. R. & Mahlous, H. Student timetabling genetic algorithm accounting for student preferences. *PeerJ Comput. Sci.* **9**, e1200 (2023).

[4] Burke, E. & Petrovic, S. Recent research directions in automated timetabling. *Eur. J. Oper. Res.* **140**, 266–280 (2002).

[5] Qu, R., Burke, E., McCollum, B., Merlot, L. & Lee, S. A survey of search methodologies and automated system development for examination timetabling. *J. Sched.* **12**, 55–89 (2009).

[6] Tan, J., Abdullah, S. & Kendall, G. A survey of the university course timetabling problem. *Ann. Oper. Res.* **239**, 293–315 (2016).

[7] Zeyad, A. A., Laabi, N. A. & Muhammad, W. S. ACO and PSO methods to solve single machine scheduling problems. *Sched. Algorithms* **1**, 37–43 (2025).

[8] Frank, I. Interactive ant colony optimization algorithms for timetabling problems. *Lect. Notes Comput. Sci.* **9835**, 19–34 (2016).

[9] Mazlan, M., Makhtar, M., Firdaus, A. & Ahmad, K. University course timetabling model using ant colony optimization algorithm approach. *Indones. J. Electr. Eng. Comput. Sci.* **13**, 72–76 (2019).

[10] Xiao, S. English teaching resources recommendation based on improved ant colony optimization algorithms. *Procedia Comput. Sci.* **228**, 464–476 (2023).

[11] Iqbal, Z., Chan, H. Y. & Ahmed, N. Effective solution of university course timetabling using particle swarm optimizer based hyper-heuristic approach. *Appl. Soft Comput.* **109**, 107543 (2021).

[12] Tassopoulos, I. X., Iliopoulou, C. A., Katsaragakis, I. V. & Beligiannis, G. N. An effective local particle swarm optimization-based algorithm for solving the school timetabling problem. *Appl. Soft Comput.* **138**, 110233 (2023).

[13] Thepphakorn, T., Sooncharoen, S. & Pongcharoen, P. Particle swarm optimisation variants and its hybridisation ratios for generating cost-effective educational course timetables. *SN Comput. Sci.* **2**, 1–12 (2021).

[14] Chen, M., Sze, S., Goh, S., Sabar, N., Kendall, G. & Abdullah, S. A survey of university course timetabling problem: Perspectives, trends and opportunities. *IEEE Access* **9**, 106515–106529 (2021).

[15] Combrink, H. M. & Marivate, V. Reinforcement learning in education: A multi-armed bandit approach. *Comput. Educ. Artif. Intell.* **3**, 100056 (2022).

[16] Alhussan, A. R., Al-Hagery, M. A. & Alqahtani, S. M. A hybrid metaheuristic approach for solving the university course timetabling problem. *Algorithms* **18**, 158 (2025).

[17] Ahmed, M., Abdullah, S. & Turabieh, H. Hybrid metaheuristic optimization for educational timetabling problems. *Comput. Ind. Eng.* **191**, 109985 (2024).

[18] Abdullah, S. & Turabieh, H. A survey of hyper-heuristic approaches for university timetabling. *Future Gener. Comput. Syst.* **146**, 89–104 (2023).

[19] Bouneffouf, D. & Rish, I. A survey on practical applications of multi-armed and contextual bandits. *ACM Comput. Surv.* **49**, 1–36 (2016).

[20] Lagos, F. & Pereira, J. Multi-armed bandit-based hyper-heuristics for combinatorial optimization problems. *Eur. J. Oper. Res.* **312**, 70–91 (2024).

[21] Meidani, K., Mirjalili, S. & Barati, A. M. MAB-OS: Multi-armed bandits metaheuristic optimizer selection. *Appl. Soft Comput.* **128**, 109452 (2022).

[22] Bashab, A. et al. Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. *Comput. Mater. Continua* **75**, 3099–3123 (2023).

[23] Chen, R. & Shih, H. Solving university course timetabling problems using constriction particle swarm optimization with local search. *Algorithms* **6**, 227–244 (2013).

[24] Alsmadi, M. K. et al. An enhanced particle swarm optimization for ITC2021 sports timetabling. *Comput. Mater. Continua* **73**, 1–18 (2022).

[25] Ahmed, A. K. S. & Mahmud, M. S. Optimization techniques for university course timetabling: A comprehensive review. *J. Sched.* **27**, 1–21 (2024).

[26] Van Parys, B. *Optimal learning for structured bandits*. (Massachusetts Institute of Technology, Cambridge, 2018).

[27] Kakarash, Z. A., Karim, S. H. T., Ahmed, N. F. & Omar, G. A. New topology control based on ant colony algorithm in optimization of wireless sensor network. *Passer Journal of Basic and Applied Sciences* **3**, 123–129 (2021).

[28] Mansour, M. M. & Al-hamdani, K. S. M. Tabu search algorithm to optimize layout design for a multi objective plant function. *Passer Journal of Basic and Applied Sciences* **6**, 446–452 (2024).

[29] Hommadi, A. H., Abass, A. A., Al-Fawzy, A. M., Al-Mohammed, F. M., Al-Ansari, N. A. & Hussien, H. H. Scheduling the laterals of Shattulhilla River by utilizing genetic algorithm as water sustainability technique. *Passer Journal of Basic and Applied Sciences* **6**, 84–93 (2024).

[30] Ahmad, R., Fattah, M., Taha, Z., Khurshid, S., Ismael, R. & Abdul, S. Calibration of the SPAW model for estimating plant available water in the Kurdistan region of Iraq. *Passer Journal of Basic and Applied Sciences* **7**, 318–324 (2025).

[31] Ramdania, D., Irfan, M. & Alfarisi, F. Comparison of genetic algorithms and particle swarm optimization algorithms in course scheduling. *J. Phys. Conf. Ser.* **1402**, 022079 (2019).

[32] Kuleshov, V. & Precup, D. Algorithms for the multi-armed bandit problem. *J. Mach. Learn. Res.* **15**, 1–48 (2014).

[33] Chen, Y., Cuellar, A., Luo, H., Modi, J., Nemlekar, H. & Nikolaidis, S. Fair contextual multi-armed bandits: Theory and experiments. *Proc. Mach. Learn. Res.* **124**, 1–18 (2020).

[34] Gao, G. et al. Combination of auction theory and multi-armed bandits: Model, algorithm, and application. *IEEE Trans. Mob. Comput.* **21**, 3562–3575 (2022).

[35] Silva, N., Werneck, H., Silva, T., Pereira, A. C. M. & Rocha, L. Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Syst. Appl.* **197**, 116669 (2022).

[36] Techie-Menson, H. & Nyagorme, P. Design and implementation of a web-based timetable system for higher education institutions. *Int. J. Comput. Appl.* **183**, 1–13 (2021).

[37] Lin, B. Reinforcement learning and bandits for speech and language processing: Tutorial, review and outlook. *Expert Syst. Appl.* **238**, 122254 (2024).