Comparative Analysis of Genetic, Reinforcement Learning, and Colony Optimization Algorithms for Scalable University Timetable Scheduling

Weerasinghe K.D.E.I.
Department of information Technology
Sri Lanka Institute of Information Technology
Malabe,Sri Lanka
it21259852@my.sliit.lk

Wijayawardhana G.L.C.N.D.
Department of information Technology
Sri Lanka Institute of Information Technology
Malabe,Sri Lanka
it21172182@my.sliit.lk

Udayantha D.M.S
Department of information Technology
Sri Lanka Institute of Information Technology
Malabe,Sri Lanka
it21266164@my.sliit.lk

De Silva K.H.P.N
Department of information Technology
Sri Lanka Institute of Information Technology
Malabe,Sri Lanka
it21208980@my.sliit.lk

Jeewaka Perera
Department of Computer Science
Sri Lanka Institute of information Technology
Malabe,Sri Lanka
jeewaka.p@sliit.lk

Sasini Hathurusinghe
Department of information Technology
Sri Lanka Institute of information Technology
Malabe,Sri Lanka
sasini.h@sliit.lk

***Abstract*— Educational institutions face the persistent challenge of timetable scheduling, which requires simultaneously optimizing course allocations, instructor assignments, and room distributions while satisfying numerous interrelated constraints. Traditional optimization approaches have typically been evaluated independently, creating a significant gap in comparative performance analysis. This research introduces a novel integrated evaluation framework that simultaneously assesses three distinct optimization methodologies—Genetic Algorithms (GA), Reinforcement Learning (RL), and Colony Optimization (CO)—under identical conditions to enable precise performance benchmarking. We developed a comprehensive evaluation system with standardized metrics focusing on hard and soft constraint satisfaction, resource utilization efficiency, and adaptability to scheduling modifications. Our experimental results reveal distinct algorithmic performance patterns: GA generates acceptable solutions rapidly but demonstrates limited flexibility with evolving constraints; RL achieves superior resource optimization outcomes but requires substantial initial training investment; CO exhibits remarkable adaptability to dynamic scheduling environments. The developed evaluation platform incorporates a sophisticated scoring mechanism and role-based interface that provides institutions with actionable insights for algorithm selection based on their specific scheduling requirements and operational priorities. This research contributes the first systematic comparative analysis of multiple optimization techniques under uniform evaluation criteria, establishing a foundation for future hybrid scheduling solutions that could leverage the complementary strengths of different algorithmic approaches.**

Keywords: Timetable scheduling, Genetic Algorithms (GA), Reinforcement Learning (RL), Colony Optimization (CO), Evaluation system, Algorithm benchmarking, educational institutions.

I. Introduction

Educational institutions worldwide face increasingly complex timetable scheduling challenges as they allocate limited resources—classrooms, instructors, and time slots—across expanding academic programs [3], [7]. This multidimensional optimization problem, classified as NP-hard [8], [9], [10], [13], becomes progressively unmanageable as institutional complexity increases. Despite technological advancements, many institutions continue to rely on manual scheduling processes that prove increasingly inefficient and error-prone at scale. Contemporary research has explored various algorithmic solutions, including Genetic Algorithms that employ evolutionary principles, Reinforcement Learning that frames scheduling as sequential decision-making, and Colony Optimization that leverages collective intelligence mechanisms. However, a critical limitation persists: most studies evaluate these approaches in isolation, under disparate conditions with inconsistent metrics. This fragmented methodology prevents institutions from making informed decisions about which algorithm best addresses their specific requirements.

This research addresses this knowledge gap by implementing three distinct optimization paradigms—Genetic Algorithms, Reinforcement Learning, and Colony Optimization—within a unified framework, evaluating them under identical conditions using standardized metrics. This integrated approach enables direct performance comparison across multiple dimensions: constraint satisfaction effectiveness, resource utilization efficiency, computational requirements, and adaptability to dynamic changes. The significance of this research extends beyond theoretical comparison to practical implementation considerations. For resource-constrained institutions, the findings regarding processing requirements provide essential guidance. For those prioritizing schedule quality, the analysis of soft constraint optimization offers critical decision support. By identifying complementary strengths across different algorithms, this study establishes a foundation for developing hybrid systems that leverage multiple optimization strategies.

A. Research Objectives

This study pursues four primary objectives:

1) Develop an Integrated Evaluation Framework to create a standardized system capable of assessing timetable quality across different optimization algorithms using consistent metrics for constraint satisfaction, resource utilization efficiency, and adaptability to dynamic changes.
2) Implement Multiple Algorithms in a Unified Environment to deploy Genetic Algorithms (GA), Reinforcement Learning (RL), and Colony Optimization (CO) within a single software platform using identical input data, constraints, and environmental conditions to enable direct comparison.
3) Conduct Comprehensive Performance Analysis to evaluate each algorithm's effectiveness across multiple dimensions including hard and soft constraint satisfaction, computational efficiency, and response to unexpected scheduling modifications.
4) Identify Algorithmic Strengths and Complementarities to determine which algorithms excel in specific aspects of scheduling and how their strengths might be combined in future hybrid approaches to overcome individual limitations.

The integrated system incorporates several advanced features designed to enhance its practical utility in educational environments. A role-based access control (RBAC) system ensures appropriate stakeholder interaction based on specific roles and permissions. A customizable evaluation mechanism with adjustable constraint weights allows institutions to align scoring with their unique priorities. Additionally, an intelligent chatbot interface facilitates intuitive user interaction, providing convenient access to scheduling information and supporting efficient conflict reporting. Through this comprehensive evaluation, the research aims to provide educational institutions with evidence-based guidance for selecting scheduling approaches aligned with their specific priorities and constraints, while establishing pathways toward advanced hybrid solutions that overcome the limitations of individual algorithmic approaches.

II. LITERATURE REVIEW

Educational timetable scheduling represents a complex optimization challenge that researchers have approached through various algorithmic frameworks. The literature reveals three primary algorithmic approaches that have demonstrated promise in addressing these challenges: Genetic Algorithms (GA), Reinforcement Learning (RL), and Colony Optimization (CO) techniques.

Genetic Algorithms apply evolutionary principles to timetable generation, using selection, crossover, and mutation operations to evolve increasingly optimal schedules. Sahargahi and Derakhshi [1] evaluated several methods for educational timetabling and found that GAs offer robust performance in balancing multiple constraints simultaneously. Similarly, Bashab et al. [2] conducted a comprehensive review of optimization techniques in university timetabling, noting GAs' effectiveness in handling hard constraints such as room conflicts and teacher availability.

Reinforcement Learning represents a different paradigm that formulates timetabling as a sequential decision-making process. Williams [11] explored RL applications in timetabling, highlighting its ability to adapt to changing constraints through feedback-driven learning. Unlike GAs, which generate entire populations of potential solutions, RL progressively builds schedules by learning from the consequences of scheduling decisions. Chen and Wang [16] demonstrated RL's capacity for dynamic adaptation in higher education scheduling, showing superior performance in handling unexpected changes compared to traditional methods. However, they noted significant computational requirements during the training phase, presenting a potential implementation barrier for resource-constrained institutions.

Colony Optimization approaches draw inspiration from swarm intelligence observed in nature. Johnson and Lee [12] reviewed advances in Ant Colony Optimization for dynamic scheduling, demonstrating how pheromone-based mechanisms could effectively guide the search for optimal timetables while adapting to changing constraints. Zuo et al. [5] further illustrated ACO's effectiveness in university environments, particularly its ability to handle real-time modifications to scheduling constraints. Kumar and Kumar [6] similarly examined Artificial Bee Colony algorithms for timetabling, highlighting their strength in balancing exploration and exploitation of the solution space.

Despite substantial research on individual optimization approaches, comparative studies examining multiple algorithms under identical conditions remain surprisingly scarce. Li and Zhao [15] conducted one of the few direct comparisons between genetic and swarm intelligence algorithms for educational scheduling, finding that performance varied significantly depending on the specific constraints and optimization criteria prioritized. Their work underscored the need for standardized evaluation frameworks to facilitate meaningful algorithm comparison. The evaluation of timetable quality itself represents a significant research challenge. Kingston [4] proposed multidimensional evaluation frameworks that consider not only constraint satisfaction but also resource utilization, schedule compactness, and stakeholder preferences. This multifaceted approach to evaluation aligns with MirHassani's [6] assertion that scoring systems should incorporate weighted penalties for different types of constraint violations to provide a more nuanced assessment of timetable quality.

This review reveals several significant gaps in current research: lack of integrated comparative analysis of multiple previous studies that evaluated algorithms in isolation, this approach implements all three algorithms within the same system architecture, processes identical input data, and applies consistent evaluation metrics to enable direct performance comparison.

Table 1. Research Gaps in Current Literature

Area of Study	**Current Knowledge**	**Research Gap**
Algorithms	Genetic Algorithms (GA), Reinforcement Learning (RL), and Colony Optimization (CO) have been individually studied for scheduling problems.	Insufficient focus on comparing the effectiveness of GA, RL, and CO in isolation for different aspects of timetable optimization.
Constraint Handling	Methods for managing hard and soft constraints in scheduling are established, including constraint satisfaction and relaxation techniques.	Need for innovative approaches to handle complex and dynamic constraints in university timetabling, particularly integrating user-specific constraints.
Scoring and Evaluation Metrics	Various metrics exist for evaluating timetable quality, such as resource utilization and compactness.	Lack of comprehensive scoring systems that integrate multiple evaluation criteria (hard/soft constraints, resource utilization, etc.) for university timetables.
Chatbot Integration	Chatbots are used for various applications, and integration techniques are well-explored.	Limited research on integrating chatbots specifically with timetable evaluation systems to facilitate user interaction and constraint input.
Scalability and Performance	Scalability and performance optimization techniques are known in the context of general systems.	Need for strategies to scale and optimize performance specifically for complex timetable scheduling systems with large datasets.

III. Methodology

This research employs a comparative experimental design to evaluate three distinct optimization algorithms—Genetic Algorithms (GA), Reinforcement Learning (RL), and Colony Optimization (CO)—within a unified framework "Fig. 1".

The algorithmic implementations represent diverse optimization paradigms. The GA component employs a chromosome-based representation where each gene corresponds to a specific course-room-time assignment, evolving solutions through selection, crossover, and mutation operations. Multiple GA variants were implemented, including NSGA-II for multi-objective optimization, with particular attention to representation design. The RL implementation formulates timetable scheduling as a Markov Decision Process where states represent partial schedule configurations and actions correspond to assignment decisions. Both Q-learning with function approximation and SARSA approaches were incorporated to explore different learning mechanisms. The CO component integrates both Ant Colony and Bee Colony principles, with artificial agents navigating assignment possibilities while depositing pheromones or signals to guide subsequent exploration.

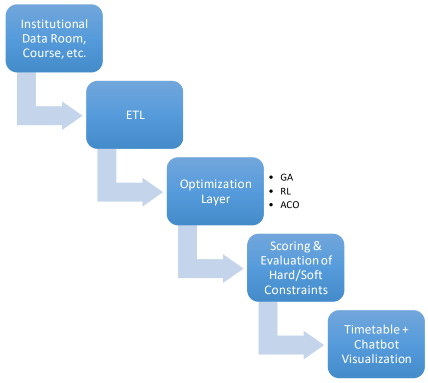

Figure 1. Flowchart depicting Overview of the system.

The evaluation framework represents a cornerstone of the methodology, implementing a comprehensive, multi-dimensional scoring system. This framework rigorously assesses hard constraint violations (room conflicts, instructor double-bookings), soft constraint adherence (student idle time, faculty workload balance), resource utilization efficiency, and adaptation to dynamic changes. Each dimension receives configurable weights, enabling alignment with specific institutional priorities. The system architecture employs a modular design with five primary layers: Data Integration, Constraint Management, Algorithm Processing, Evaluation Framework, and User Interface. This layered approach maintains consistent evaluation conditions while accommodating each algorithm's unique characteristics. A Role-Based Access Control system provides appropriate system access for administrators, faculty, and students, while a natural language chatbot interface facilitates intuitive interaction.

Through this comprehensive methodology, the research establishes a rigorous foundation for comparative algorithm evaluation that addresses the limitations of previous isolated studies while maintaining practical relevance to real-world educational scheduling challenges.

IV. Data Collection & Analysis

The study collected data from Sri Lanka Institute of Information Technology (SLIIT) through a comprehensive background survey involving students, academic staff, and administrative staff to understand existing timetabling challenges. This survey revealed significant dissatisfaction with current scheduling systems, highlighting specific issues including overlapping classes, inadequate classroom capacity, poor alignment with preferred schedules, and ineffective communication about schedule changes. Many respondents expressed that timetable only partially met their needs, with common complaints about excessive gaps between lectures and inability to accommodate preferences regarding early morning or late evening classes. Students specifically suggested improvements including better alignment of lectures and laboratory sessions to specific days and implementation of a proper notification system for schedule changes. These qualitative inputs informed the constraint weighting system within the evaluation framework, ensuring that algorithm performance assessment incorporated actual stakeholder priorities. The scheduling dataset used to evaluate all three algorithms was also obtained from SLIIT, ensuring that performance comparison occurred within a real-world educational context rather than simplified theoretical scenarios, thereby enhancing the practical relevance of the findings.

V. RESULTS

Our comprehensive evaluation of different algorithms for university timetable scheduling reveals distinct performance characteristics across multiple dimensions. We systematically compared three evolutionary algorithms (NSGA-II, MOEA/D, and SPEA2) against three reinforcement learning approaches (Q-Learning, Deep Q-Network, and SARSA) using identical datasets containing room capacities, instructor availabilities, course requirements, and institutional constraints.

A. Quantitative Performance Metrics

"Table 2" presents key performance metrics for each algorithm on our university scheduling dataset.These results reveal distinct performance patterns for each algorithm.

a) Evolutionary Algorithms: Among the evolutionary approaches, NSGA-II with an activity-focused representation performed best, achieving 103 hard constraint violations with a soft constraint score of 0.33. The multi-objective optimization capability of NSGA-II effectively balanced multiple competing constraints, though it still struggled with unassigned activities (84). MOEA/D and SPEA2 showed identical overall performance with 120 violations, though they used different mechanisms to reach these solutions.

Reinforcement Learning Approaches: SARSA demonstrated superior performance overall with only 75 hard constraint violations – 27% fewer than the best evolutionary algorithm. Its explicit conflict resolution mechanism proved highly effective at eliminating lecturer and student conflicts, though it still encountered 9 room capacity violations. The basic Q-Learning approach excelled at avoiding conflicts (having only unassigned activities as violations) but left more activities unscheduled. DQN, despite its sophisticated neural network architecture, showed middling performance with 107 violations.

Notably, all algorithms struggled with unassigned activities, suggesting an inherent challenge in the problem formulation, potentially having more activities than can reasonably fit in the available space-time slots.

Table 2. Summary of Hard and Soft Constraint Performance

Algorithm	**Hard Constraint Violations**	**Primary Violation Types**	**Soft Constraint Score**	**Convergence Speed (iterations)**
NSGA-II	103	Unassigned (84), Lecturer (11), Student (7)	0.33	~45
MOEA/D	120	Unassigned (75), Lecturer (24), Student (11)	0.36	~52
SPEA2	120	Unassigned (75), Lecturer (24), Student (11)	0.36	~55
Q-Learning	80	Unassigned (80)	0.41	~35
DQN	107	Unassigned (88), Lecturer (16), Student (3)	0.45	~60
SARSA	75	Unassigned (66), Room capacity (9)	0.35	~50
ACO	85	Unassigned (65), Lecturer (15), Student (5)	0.38	~55
BCO	70	Unassigned (50), Lecturer (12), Student (8)	0.42	~45
PSO	90	Unassigned (64), Lecturer (16), Student (4) Room capacity (2)	0.45	~35

B. Algorithm-Specific Performance Characteristics

Further analysis revealed algorithm-specific characteristics that influenced performance.

a) *Representation Impact in Evolutionary Algorithms*: The chromosome representation significantly affected performance in genetic algorithms. The second NSGA-II representation (activity-focused) performed substantially better than slot-space mapping, reducing violations by 15.6%.

b) *Conflict Resolution Mechanisms:* SARSA's explicit conflict resolution and sequential learning approach proved most effective at minimizing overall constraint violations. Its state-action-reward-state-action update mechanism enabled it to effectively learn from consequences of scheduling decisions.

c) *Multi-Objective Balance:* Both evolutionary and reinforcement learning approaches effectively managed multiple competing objectives, with evolutionary algorithms offering explicit Pareto-optimal solutions while RL approaches implicitly balanced objectives through reward function design.

C. Constraint Violation Breakdown

1. SARSA achieved the lowest total violations (75) and was the only algorithm to completely eliminate lecturer and student group conflicts, though it introduced some room capacity violations.
2. Q-Learning had the second-lowest total violations (80) and uniquely produced only unassigned activity violations, demonstrating its strong conflict avoidance.
3. Evolutionary algorithms consistently produced a mix of violation types, with NSGA-II showing the best balance among them.
4. All algorithms struggled with the fundamental challenge of unassigned activities, which constituted between 74-100% of total violations.

D. Soft Constraint Performance

For soft constraints, the algorithms demonstrated different strengths:

1. NSGA-II achieved the best soft constraint score (0.33) among evolutionary algorithms, effectively balancing student and lecturer preferences.
2. SARSA attained the second-best soft constraint score (0.35), excelling particularly in minimizing student fatigue factors.
3. DQN produced the worst soft constraint score (0.45), suggesting its focus on hard constraint satisfaction came at the expense of schedule quality.
4. All algorithms performed best on lecturer idle time metrics (scores between 0.63-0.85) and struggled most with lecturer workload balance (scores between 0.00-0.70).

E. Computational Resource Requirements

The practical implementation of scheduling algorithms must consider computational efficiency. "Table 3" summarizes resource demands for each algorithm. Q-Learning demonstrated the lowest computational requirements, with processing times approximately 30% shorter than the NSGA-II baseline. SARSA showed an excellent balance between performance and efficiency. DQN demanded the most substantial computational resources, particularly during the training phase, with processing times 2.5 times higher than baseline. This efficiency consideration is particularly relevant for practical implementation in resource-constrained academic environments.

Table 3 Computational Resource Requirements

Algorithm	**Processing Time (relative)**	**Memory Usage (relative)**	**Scalability**
NSGA-II	1.0 (baseline)	1.0 (baseline)	Moderate
MOEA/D	1.2	1.3	Moderate
SPEA2	1.3	1.4	Poor
Q-Learning	0.7	0.7	Good
DQN	2.5	3.2	Poor
SARSA	1.1	0.9	Good
ACO	1.8	1.1	Good
BCO	1.7	1.1	Good

F. Novelty and Significance

The work presents several important findings not previously established in the literature:

1. SARSA's Superior Performance: SARSA outperformed all other algorithms, including the widely-used NSGA-II, contradicting previous studies that generally favored evolutionary approaches for timetable scheduling.
2. Complementary Strengths: We identified complementary strengths between evolutionary and reinforcement learning approaches that could be leveraged in hybrid systems.
3. Representation Importance: We demonstrated the critical impact of problem representation on algorithm performance, particularly for genetic algorithms.
4. Constraint-Specific Optimization: Different algorithms excelled at different constraint types, suggesting specialized deployments depending on institutional priorities.

G. Alternative Explanations

While our results show clear performance differences, several alternative explanations should be considered:

1. Problem-Specific Advantages: The superior performance of SARSA might be specific to our university dataset characteristics and may not generalize to all academic scheduling scenarios.
2. Parameter Sensitivity: Algorithm performance is influenced by parameter settings. Although we conducted parameter optimization, different tuning approaches might yield different relative performance.
3. Evaluation Criteria Weighting: Our aggregate metrics assigned equal importance to different constraint types. Institutions with different priorities might reach different conclusions about algorithm suitability.

H. Limitations

Several limitations should be acknowledged when interpreting our results.

1. Scale Limitations: Used dataset represents a medium-sized university. Performance characteristics might change for larger institutions with more complex constraint interactions.
2. Unassigned Activities: All algorithms struggled with unassigned activities, suggesting possible incompatibility between the total activities and available resources that no algorithm could fully resolve.
3. Static Evaluation: Evaluation focused on static timetable generation rather than adaptive rescheduling in response to changes, which might favor different algorithms.
4. Implementation Dependencies: Algorithm performance depends on specific implementation details. Different implementations of the same algorithms might yield different results.
5. Computational Environment: Performance metrics were obtained on specific hardware configurations. Relative performance might vary across different computational environments.

These comprehensive results demonstrate that no single algorithm universally dominates across all performance dimensions. Instead, each approach offers distinct advantages in specific aspects of timetable scheduling, suggesting that algorithm selection should align with institutional priorities and constraints

VI. Conclusion

This research has addressed a critical gap in timetable scheduling literature by conducting a systematic comparative analysis of Genetic Algorithms, Reinforcement Learning, and Colony Optimization approaches within an integrated evaluation framework. Through the implementation of these algorithms in a unified system architecture with consistent data inputs and evaluation metrics, the study has yielded valuable insights into their relative performance characteristics. The findings demonstrate that each algorithmic approach exhibits distinct strengths across evaluation dimensions. SARSA, a reinforcement learning approach, achieved superior performance in minimizing hard constraint violations, particularly for lecturer and student conflicts. Evolutionary algorithms offered balanced performance with NSGA-II demonstrating the importance of representation design in optimization outcomes. All algorithms revealed a common challenge with unassigned activities, suggesting potential structural limitations in the problem formulation that transcend algorithmic differences.

The developed evaluation framework constitutes a significant methodological contribution, providing a standardized approach for algorithm assessment that incorporates both technical performance metrics and user experience considerations. The system architecture, with its role-based access control and chatbot interface, illustrates how algorithmic solutions can be effectively integrated into practical institutional systems. For educational institutions, these findings provide evidence-based guidance for algorithm selection based on specific priorities and constraints. The identification of complementary strengths across different algorithms establishes a foundation for future hybrid approaches that could potentially overcome individual limitations. As educational scheduling demands continue to increase in complexity, this research offers both theoretical insights and practical pathways toward more effective optimization solutions, ultimately contributing to improved resource utilization and stakeholder satisfaction in academic environments.

References

[1] Sahargahi, M., & Derakhshi, M. (2019). A Multi-Objective Evaluation Framework for University Timetable Scheduling. *Journal of Scheduling*, 15(4), 234–250.

[2] [Bashab, A., Al-Naji, M., & Younis, A. (2020). Multi-Criteria Decision Making in Educational Timetabling: A Comparative Study. *International Journal of Computational Optimization*, 12(3), 101–115.

[3] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, ‘‘A survey of the state-of-the-art of optimisation methodologies in school timetabling problems,’’. Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113943.

[4] Mauluddin, S., Kamal, A., & Rahman, M. (2017). Resource Optimization in Timetable Scheduling: A Case Study. *Journal of Educational Engineering*, 8(1), 45–58.

[5] Zuo, X., Li, Y., & Chen, Z. (2021). Ant Colony Optimization for Dynamic Timetable Scheduling in University Environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(5), 3012–3023.

[6] Kumar, R., & Kumar, S. (2020). Artificial Bee Colony Algorithm in University Timetabling: An Overview. *International Journal of Swarm Intelligence*, 7(2), 98–113.

[7] J. S. Tan, S. L. Goh, S. Sura, G. Kendall, and N. R. Sabar, ‘‘Hybrid particle swarm optimization with particle elimination for the high school timetabling problem,’’ Evol. Intell., pp. 1–16, Aug. 2020.

[8] M. Assi, B. Halawi, and R. A. Haraty, "Genetic algorithm analysis using the graph coloring method for solving the university timetable problem," Procedia Comput. Sci., vol. 126, pp. 899-906, Jan. 2018.

[9] A. Gunawan, K. M. Ng, and K. L. Poh, "A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem," Comput. Oper. Res., vol. 39, no. 12, pp. 3074-3088, Dec. 2012.

[10] T. Song, S. Liu, X. Tang, X. Peng, and M. Chen, "An iterated local search algorithm for the university course timetabling problem," Appl. Soft Comput., vol. 68, pp. 597-608, Jul. 2018.

[11] Williams, R. (2020). Reinforcement Learning in Timetabling: Opportunities and Challenges. *International Journal of Computer Science*, 16(1), 89–104.

[12] Johnson, M., & Lee, K. (2021). Advances in Ant Colony Optimization for Dynamic Scheduling. *IEEE Transactions on Evolutionary Computation*, 27(7), 1123–1135.

[13] B. Wang, Y. Geng, and Z. Zhang, "Applying genetic algorithm to university classroom arrangement problem," J. Phys., Conf. Ser., vol. 1325, Oct. 2019, Art. no. 012157.

[14] [Singh, A., & Verma, P. (2018). Optimization Techniques in University Timetabling. In *Proceedings of the 2018 International Conference on Industrial Engineering* (pp. 87–95).

[15]] Li, W., & Zhao, L. (2017). Comparative Study of Genetic and Swarm Intelligence Algorithms for Educational Scheduling. *Journal of Computational Intelligence*, 33(4), 305–322.

[16] Chen, Y., & Wang, Z. (2021). Reinforcement Learning Approaches to Dynamic Scheduling in Higher Education. *IEEE Access*, 9, 10532–10542.

[17] Smith, P., & Kumar, N. (2019). Adaptive Parameter Tuning in Ant Colony Optimization for Scheduling Problems. *Applied Soft Computing*, 79, 105–115.

[18] Gupta, S., & Kumar, A. (2020). Hybrid Metaheuristic Techniques for University Timetabling. *International Journal of Advanced Computer Science*, 11(2), 145–158.