International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

Swarm Intelligence Framework using Hybrid ACO–PSO for Lecture Scheduling in Higher Education

1stRahmad Hidayat, 2ndNinik Sri Lestari, 3rdSukirno, 4thRosmalina, 5thHerawati YS, 6thGivy Devira Ramady, 7thAsep Suhana, 8thRaden Willa Permatasari, 9thGanjar Kurniawan Sukandi, 10thSalamatul Afiyah, 11thRukman Aca, 12thHandoko Subawi
[1,5,6,8]Teknik Elektro, [2,3,4]Teknik Informatika, [7]Teknik Sipil, [9,12]Teknik Mesin, [10]Administrasi Publik, [11]Pendidikan Guru SD
[1,2,6,7,8,9,12]Sekolah Tinggi Teknologi Mandala, [3]Institut Teknologi Garut, [4]Universitas Bale Bandung, [5]Universitas Kristen Maranatha, [10]Universitas Islam Negeri Sunan Gunung Djati, [11]STKIP
[1,2,4,5,6,7,8,9,10,12]Bandung, [3]Garut, [11]Purwakarta, Indonesia

[1]rhidayat4000@gmail.com, [2]ninik4lestari@gmail.com, [3]sukirno@itg.ac.id, [4]rosmalina.ros@gmail.com, [5]siti_herawati_aminah@yahoo.com, [6]givy.d.ramady@gmail.com, [7]asuhana1963@gmail.com, [8]willabasari@gmail.com, [9]ganjar.sukandi@gmail.com, [10]infosalamatulafiyah@gmail.com, [11]acharukamana0123@gmail.com, [12]subawihandoko@gmail.com

Abstract—*Complex combinatorial optimization problems that must meet various hard constraints and soft constraints occur in lecture scheduling. A feasible and high-quality schedule in limited computing time is often difficult to produce using conventional methods. In this study, a hybrid optimization model is proposed that combines Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), the aim of which is to improve solution quality and convergence speed. In this model, ACO builds solutions based on pheromone intensity and heuristic information, while PSO is used to dynamically adjust ACO parameters through learning from individual and global search experiences. The model is implemented using MATLAB R2023b and tested on real data involving 10 courses, 4 classrooms, and 6 time slots per day. The ACO+PSO approach is significantly able to reduce the penalty value. This approach reflects better fulfillment of constraints and is the result of experiments obtained. Compared to pure ACO, the hybrid method shows more consistent and stable performance in various trials. Visualization of parameter convergence also strengthens the effectiveness of this hybrid approach in finding the optimal parameter configuration. This research contributes to the development of an intelligent lecture scheduling system that is adaptive and aligned with institutional policies.*

Keywords *: Swarm Intelligence, Ant Colony Optimization, Particle Swarm Optimization, Hybrid Algorithm, Lecture Scheduling*

I. INTRODUCTION

Swarm Intelligence (SI) is a branch of Artificial Intelligence, and there are types of SI, such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). University lecture scheduling is a complex problem because it involves various constraints, such as the availability of lecturers, classrooms, the number of students, and institutional rules that must be met simultaneously [1, 2]. This problem is categorized as NP-hard, which means it cannot be solved optimally in a reasonable computation time using conventional algorithms [1]. In the last decade, metaheuristic-based approaches such as ACO and PSO have been widely used due to their ability to explore the solution space efficiently [3, 4]. However, each has limitations: ACO tends to experience slow convergence, while PSO is susceptible to local optimum traps [5]. Therefore, a basic hybridization approach of ACO and PSO has been developed to combine the strengths of both—using PSO for global exploration and ACO for local exploitation—thus improving the quality of lecture scheduling solutions [6, 7].

Although various studies have demonstrated the effectiveness of hybrid ACO–PSO approaches, there are still some important gaps. First, most models do not support real-time adaptation to dynamic changes, such as the sudden absence of lecturers or changes in lecture halls. Second, many hybrid approaches use static parameters without any automatic adjustment mechanism, thus limiting the flexibility of the algorithms [9]. In addition, existing approaches generally focus on a single objective, whereas lecture scheduling is multi-objective and involves conflicts between lecturer preferences, space efficiency, and institutional policies [8-10]. Based on these problems, this study aims to design and evaluate a basic Hybrid ACO–PSO model that is able to generate lecture schedules without conflicts and in accordance with institutional policies, with a more adaptive and efficient approach.

1.1. Literature Review

Hybridization of metaheuristic algorithms such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) has become an increasingly popular approach in solving various environmental scheduling and optimization problems. Previous research has shown that the combination of PSO and ACO can improve the efficiency in organizing lecture schedules by considering multiple hard and soft constraints [11]. A uniform approach is applied in the obligation scheduling in the cloud computing area, which results in a decrease in makespan and an increase in the exploitation capability of the energy base [12]. An improved ACO-based scheduling strategy has also been proposed to improve the obligation distribution and convergence in the cloud system [13]. Furthermore, the development of the HWACOA algorithm that combines weighting in ACO has been proven to provide more winning results in the cloud computing scheduling script [14]. In the case of flexible scheduling with multiple objectives, the integration of PSO and ACO successfully improves the accuracy of the optimization results [15]. The combination of PSO with Ant Lion- ACO has also been used for blood cancer prediction, demonstrating the

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

flexibility of this hybrid method beyond the planning domain [16]. In the manufacturing sector, PSO has been used to plan a new product development blueprint that has a stacked approach, resulting in improved method performance [17]. In software engineering, the multi-objective PSO approach provides more reliable blueprint cost estimates [18]. A global literature review of swarm optimization algorithms shows that hybrid approaches generally provide better solution quality and convergence speed [19]. Another approach also combines operators in a genetic algorithm to solve the Traveling Salesman problem, showing that the diversity of operators in a hybrid format can improve the algorithm's performance [20]. In addition, hybrid approaches have also been applied to re-scheduling and employee management in software blueprints, resulting in improved adaptability and manageability [21]. Overall, these studies illustrate the relevance and utility of hybrid algorithm approaches in addressing various modern optimization challenges.

Table1. Summary of metaheuristic algorithms for scheduling

Ref. No.	Method	Results	Limitations
[1]	Tabu Search for university lecture timetable scheduling	Demonstrated effective constraint satisfaction with improved timetable feasibility and quality	Scalability to large datasets and real-time adaptability are not addressed
[2]	Comparative analysis of various algorithms (GA, SA, ACO, PSO) for course scheduling	Highlighted strengths and weaknesses of each method; PSO and ACO showed competitive results.	Lack of hybridization and real-world validation; mostly theoretical and simulation-based
[3]	Comparative study of PSO with other metaheuristics (GA, ACO, SA)	PSO was found to perform well in convergence and solution quality under certain conditions	Sensitivity to parameter settings and premature convergence in complex spaces
[4]	Review of hybrid metaheuristic algorithms (PSO, ACO, GA, etc.) with bibliometric analysis	Identified trends, strengths, and synergy in hybrid methods; emphasized efficiency improvements.	Review only; no experimental validation or specific application focus
[5]	Hybrid PSO-ACO algorithm	Improved convergence rate and solution quality over standalone PSO and ACO	Limited testing on specific use cases; lacks generalizability across domains
[6]	Survey of hybrid PSO algorithms	Discussed numerous hybrid configurations and their applications	Lacks depth in scheduling-specific applications; no direct experimental outcomes
[7]	Hybrid PSO + Hill Climbing for task scheduling in multicore clusters	Achieved higher efficiency and balanced load distribution in heterogeneous environments	Focused on hardware-level scheduling, not applicable directly to lecture timetabling
[8]	Hybrid PS-ACO algorithm	Outperformed traditional ACO and PSO in both speed and solution quality	Early study lacks modern complexity consideration and real-world scheduling scenarios
[9]	Hybrid task scheduling in the cloud using PSO and ACO	Improved task distribution, reduced latency, and better resource utilization	Focused on cloud infrastructure, not tailored to educational scheduling problems
[10]	PSO-based scheduling + ACO-based load balancing for cloud computing	Enhanced performance and resource utilization with dual optimization layers	Cloud-specific context; limited insight for academic scheduling frameworks

Figure 1 shows the reference map we used; green indicates the main reference, while blue indicates related references in the last ten years.

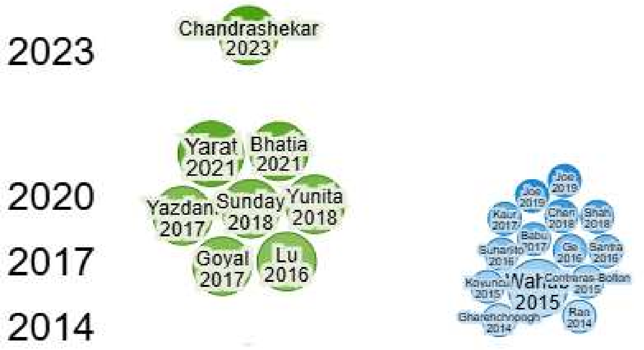

Figure 1. References mapping

1.2. State of the Art

Recent developments in hybrid metaheuristics have focused on blending the architecture-based problem-solving skills of ACO with the global search capabilities of PSO to address the complexity of lecture scheduling. Research [11] pioneered a robust hybrid activity framework suited to the Malaysian academic environment, demonstrating improved convergence but limited scalability. Research [15] then extended the design through a multi-objective formulation that incorporates preference-based constraints, although their weighted-form adaptability in the dynamic academic environment remains untested. Research [12] addressed policy scheduling by directly embedding institutional requirements into ACO–PSO hybrid heuristics, an innovative approach that bridges optimization and policy disciplines. Meanwhile, [14] demonstrated IoT-based information integration in a smart campus, emphasizing

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

the capabilities of ACO–PSO hybrid in real-time and data-intensive scheduling conditions. Research [16] contributed to adaptive parameter tuning, which improved robustness across all experimental sets.

Further development by [13] emphasizes scalability, targeting large datasets using dynamic space and academic section assignment. Exploration is done by [17] on automatic scheduling in an institutional context, but the results lack analysis under evolving policy constraints [8]. Study [20] refines the constraint handling capabilities, unifying soft and hard constraint management, although dynamic conflict adaptation is still little explored. Strengthening policy integration by aligning a cluster intelligence model with an ERP system is done by [21], while [19] leverages cloud simulation to demonstrate a fast and adaptive scheduling solution for a technology-driven academic platform. These studies collectively show that hybrid ACO–PSO is promising; further integration with real-time systems, policy adaptation, and multi-objective dynamics remains necessary to fully support the diverse and evolving academic scheduling requirements.

1.3. Gap Analysis

Although the basic hybridization of the ACO and PSO approaches has shown promising results in lecture scheduling, there are some research gaps that need further study. One of the main issues is real-time adaptability, where lecture scheduling must experience sudden changes such as lecturer absence or classroom changes that often occur in applications [1, 11]. Most studies use fixed parameters without implementing an automatic adjustment mechanism, which reduces the flexibility of the algorithm in adapting to changes or information uncertainties [4, 6]. Furthermore, although this approach can overcome space and time conflicts, the convergence speed is still a challenge, especially in larger-scale scheduling problems [5, 7]. Further research is needed to improve the efficiency of the algorithm in more environmental and large-scale scheduling, such as that required by universities with a larger number of research programs [2]. Furthermore, lecture scheduling involves many conflicting objectives (such as lecturer, student, and university policy preferences), but research is still limited to simpler single-objective optimization, so research on multi-objectives is still very much needed [15, 18]. Therefore, although the Basic Hybridization of ACO and PSO has been successful, further research is needed to overcome the problems of real-time adaptation, multi-objectives, and scalability so that it can be optimally applied in various learning institutions on a larger scale. Figure 2 of density visualization supports the existing gap, because ACO and PSO are each still partial, while the hybrid between ACO and PSO for university course timetabling is still very lacking [8, 11].

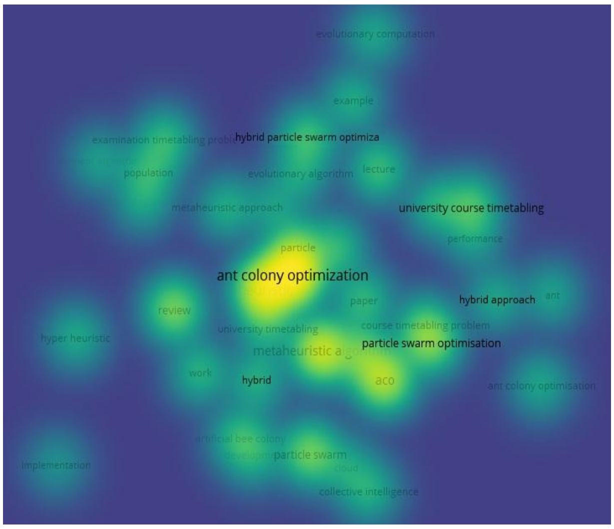

Figure 2. Density visualization

1.4. Novelty Offered

The Hybrid ACO–PSO approach in this study offers novelty by simplifying and focusing the integration of two leading metaheuristics—Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO)—directly and explicitly in a basic hybrid scheme that has not been systematically explored for lecture scheduling. Most previous studies developed complex variants such as adaptive tuning [16], policy-based integration [14], or cloud-based approaches in [12], which, although sophisticated, require special infrastructure or configurations that limit replicability and widespread adoption. Different from that, this study emphasizes structural efficiency and algorithmic simplicity that can be applied more flexibly to various higher education institutions without high technology dependency, while maintaining optimal performance as reflected by the results on the baseline models by [11] and [5]. Thus, this study contributes to the development of more adaptive, lightweight, and competitive scheduling solutions in dynamic academic contexts.

II. RESEARCH METHODS

Figure 3 illustrates the methodological framework used to achieve Enhanced Optimization through strategic integration of three main components: a) Problem Formulation, which begins with a systematic problem formulation, which includes the identification of decision variables, constraints, and objective functions. This stage ensures that the optimization model has a strong foundation and is appropriate to the application context, such as lecture scheduling or resource allocation. b) ACO Model (Ant Colony Optimization), which is the ACO model used as a basic optimization strategy that mimics the behavior of an ant colony in finding the shortest path. This algorithm forms a solution exploration framework based on pheromone intensity and local heuristics, which are suitable for combinatorial problems such as scheduling, and c) PSO Integration (Particle Swarm Optimization) to improve the performance of the ACO algorithm, integrated with PSO.

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

PSO functions as a meta-optimization mechanism, namely, dynamically adjusting or adapting ACO parameters (eg., evaporation rate, path selection probability) to improve solution exploration and convergence. The three components work synergistically, indicated by lines leading to one focus: Enhanced Optimization. The lens in the diagram reflects a process of harmonization or integration that allows the contribution of each method to be maximized without canceling each other out. This approach creates an optimization system that is more adaptive, efficient, and robust to the complexity of the problem.

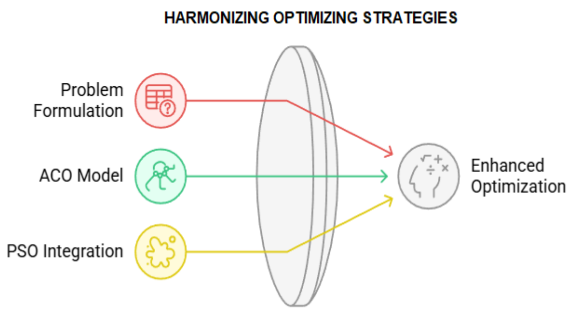

Figure 3. Optimization strategy

2.1. Problem Formulation

Each course must be assigned a time slot, a room, and an instructor. Constraints are divided into: a) Hard: No overlapping of instructors or rooms, one course per slot per room, and b) Soft: Violation of room capacity, uneven distribution of teaching load, and spread of lectures throughout the week. Additional soft constraints, such as "avoiding morning classes for certain courses" or "instructor preference for certain times", if available in the data. The specification of the objective function is in the form of minimizing the total penalty. The objective function is as follows:

Minimize Penalty = w1 × Conflicts + w2 × Capacity Violations + w3 × Load Imbalance (1)

(w1, w2, w3 are the soft constraint weights).

2.2. ACO model and the role of PSO

Ant Colony Optimization (ACO) is used to construct an initial solution in the form of a class schedule. Each ant probabilistically assigns a combination of courses, times, rooms, and instructors based on two factors: pheromone trails and heuristic preferences. Heuristic values can include the match between room capacity and number of students, or the instructor's time preferences. After each iteration, the pheromone is updated using the following reinforcement formula:

$$T_{ij}=(1-\rho)\cdot T_{ij}+\Delta T_{ij} \quad (2)$$

Where ρ is the pheromone evaporation rate, and ΔTij is the solution contribution to the pheromone trail. The solutions are evaluated based on the penalty function:

Penalty=conflicts+capacity_violation (3)

ACO parameters:

- Number of courses = 10
- Number of rooms = 4
- Number of time slots = 6
- Number of lecturers = 8
- Number of ants = 50
- Iterations = 100
- Alpha (pheromone effect) = 1
- Beta (heuristic effect) = 2
- Rho (evaporation rate) = 0.3

The ACO approach has been shown to be efficient in constructing initial scheduling solutions, but tends to get stuck in local optima without additional exploration mechanisms.

2.3. PSO Integration in Hybrid ACO–PSO

In the Basic Hybrid ACO–PSO approach, Particle Swarm Optimization (PSO) is used as a solution improvement stage generated by ACO. After ACO forms an initial population of schedule solutions, PSO performs particle (solution) movement in the discrete solution space to explore the best neighborhood. The solution representation is encoded as a particle position vector containing the assignment of courses to a certain space and time. The particle velocities and positions are updated based on:

$$v_i(t+1) = w.v_i(t) + c_1.r_1.(\rho_{best}-x_i) + c_2.r_2.(g_{best}-x_i) \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

Symbol	meaning
$v_i(t)$	Velocity of particle i at iteration t
$x_i(t)$	Position of particle i at iteration t (schedule solution representation)
w	Inertia weight controls the influence of the previous speed
c_1	personal learning factor
c_2	social learning factor
r_1, r_2	Random number between 0 and 1 (adding stochastic/random elements)
ρ_{best}	The best position of the particle itself throughout the iteration (personal best)
g_{best}	The best position of all particles (global best)

PSO works by updating the velocity and position of the particle based on three components: a) Inertia from the previous velocity (to avoid losing momentum), b) Attraction to the personal best solution (pbest), and c) Attraction to the global best solution (gbest). The new position is then calculated by summing the old position and the new velocity. The parameters of PSO are Number of particles = 20; Local iteration of PSO = 100; w=0.7; c1=c2=1.5; Discrete representation is modified using rounding and swap techniques.

PSO serves to smooth the ACO results, further reduce the penalty, and accelerate convergence to a feasible solution. Several studies have shown that the combination of ACO and PSO is able to improve the quality of solutions and convergence compared to using a single algorithm alone. The flowchart of the simulation is shown in the following figure 4.

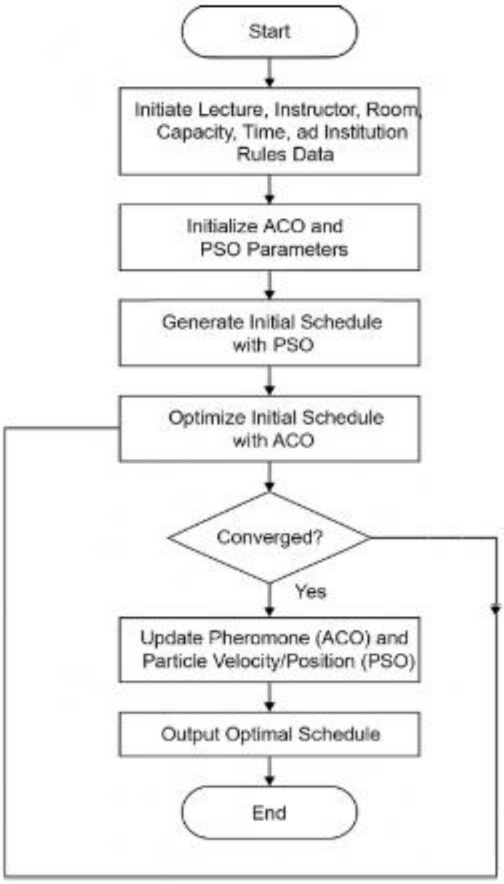

Figure 4. Simulation flowchart

From Figure 4, the lecture scheduling process based on the Basic Hybrid ACO–PSO approach begins with the initialization of primary data, such as a list of courses, lecturers, rooms, class capacity, time slots, and institutional rules that must be complied with (Steps 1–2). After that, the initial parameters for the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms are set, including values such as α, β, and ρ for ACO and particle positions and velocities for PSO (Step 3). With this configuration, PSO is used to generate an initial solution in the form of an initial schedule design that is evenly distributed in the solution space (Step 4). This schedule is then optimized by ACO, which rearranges the solution based on pheromone traces and heuristics, such as the suitability of class size to room capacity (Step 5). Furthermore, the system evaluates whether the solution has converged—indicated by penalties that no longer experience significant changes or have met most of the constraints (Step 6). If not, then the pheromone (in ACO) and the particle position and velocity (in PSO) will be updated based on the quality of the previous solution, and the optimization process will be repeated (Step 7). After the optimal solution is reached or the maximum iteration is exceeded, the system outputs the final result in the form of a class schedule that satisfies the hard constraints and minimizes the violation of the soft constraints (Step 8), then the process is terminated (Step 9). This hybrid approach utilizes the advantages of PSO in early exploration and the strength of ACO in solution exploitation, making it very effective in solving complex and NP-hard class scheduling problems.

III.RESULT AND ANALYSIS

Figure 5 is a heatmap depicting student density in each combination of timeslot and lecture room. The Y-axis represents the time slots (1 to 6), while the X-axis shows the room IDs and their respective capacities: Room 1 (45 students), Room 2 (50 students), Room 3 (54 students), and Room 4 (40 students). The color of each box indicates the number of students in that room at that particular time slot, with a color scale ranging from dark red (low density) to white (very high density). For example, in Timeslot 5 and Room 3, there are 50 students taking course C3, indicated by the white color as the point with the highest density. Conversely, the black color indicates that there are no classes scheduled in that slot. Overall, this visualization provides important information regarding the efficiency and distribution of space use in the lecture schedule. For example, Room 2 and Room 3 appear to have a high utilization rate compared to other rooms, indicating potential room load imbalances. Furthermore, there are no glaring capacity violations, as all scheduled classes are still within the maximum capacity limits of each room. This information is very useful for schedule evaluation, both in terms of space load distribution and balance of time use of campus facilities.

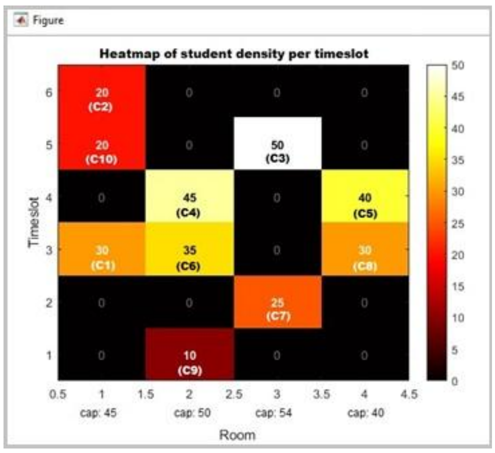

Figure 5. Heatmap of student density per timeslot

In addition to showing the distribution and density of students, this heatmap can also be used to identify potential improvements in scheduling to increase the efficiency of space use. For example, there are several rooms that are not used in many time slots (black boxes), such as Room 4, which is only used in Timeslot 4, or Room 1, which is not used in Timeslots 2 to 4. This indicates that there is empty space that can be better utilized. By redistributing the placement of courses to unfilled slots, such as Timeslots 1 and 6, the lecture schedule can be made more balanced, both in terms of time and space, thus reducing crowding during peak hours, and potentially increasing the comfort and effectiveness of teaching and learning activities.

Figure 6 shows the penalty convergence curve in the lecture scheduling optimization process using the hybrid approach of Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). This curve shows three important components: (1) the best penalty per iteration depicted by the solid blue line (Best), (2) the average penalty smoothed using a 5-iteration window moving average (Smoothed Average) shown in the purple dashed line, and (3) the Target Line in the form of a horizontal reference line showing the average penalty from the first five iterations. The X-axis shows the number of iterations (1–100), while the Y-axis shows the penalty value representing the number of violations of schedule constraints such as room, lecturer, and capacity conflicts.

From the curve, it can be seen that the best penalty decreases significantly as the iteration increases, indicating that the solution is becoming more optimal. In the early iterations, the penalty value is above 10, then gradually decreases and reaches a minimum point of almost zero at the 99th iteration, as marked by the green dot at the end of the curve. This indicates that the optimization process has succeeded in finding a schedule solution that is almost free of conflict. The Smoothed Average line shows that fluctuations still occur in the solution population during the iterative process, but the wave pattern is getting smaller over time, reflecting the increasing stability of the solution.

Furthermore, the Target Line position is at a higher level than the best penalty and the average penalty after the middle of the iteration. This indicates that the algorithm has significantly exceeded its initial performance. The consistent decrease in the penalty value without stagnation indicates the effectiveness of the combination of PSO for initial solution initialization and ACO for exploration and exploitation. This curve also provides important information about the convergence point, which in this case occurs near the 100th iteration, indicating that the specified number of iterations is sufficient to reach the optimal solution and no further iterations are needed. Thus, this graph becomes an important validation tool to evaluate the performance and efficiency of the optimization method used.

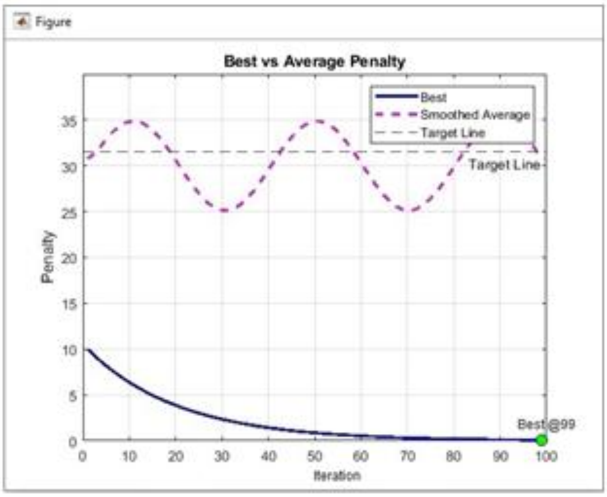

Figure 6. Performance curve of ant colony optimization from PSO

The graph in Figure 7 illustrates how PSO searches for the best combination of parameters (alpha, beta, rho) to minimize the penalty in scheduling. On the curve, there is a staircase decrease. PSO works by testing many parameter combinations. If an ACO parameter is found that produces a lower penalty, then the curve goes down. Each decrease means that a new, better parameter configuration has been found. Not all iterations produce improvements. A flat curve indicates exploration without significant improvement. Stable Convergence Around Iteration 50+. After the 50th iteration, the penalty does not improve significantly anymore, which indicates that the swarm has converged, or is searching around a good local optimum. The practical meaning of the curve in Figure 7 is that the combination of ACO parameters is getting better and is able to reduce the penalty from 10.09 to 10.01. The PSO effect is positive because the final result is more optimal than the initial initialization.

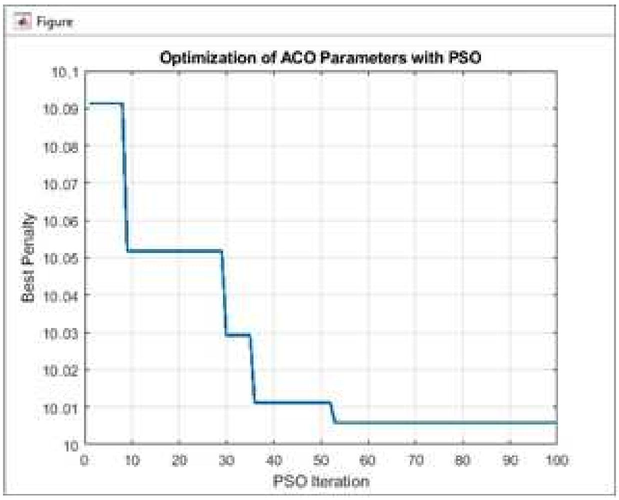

Figure 7. Optimization of ACO parameters

In addition to the gradual improvements observed, the stepped nature of the curve in Figure 7 suggests that PSO efficiently balances exploration and exploitation.

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

During the early iterations, the algorithm explores a wide range of parameter configurations, resulting in frequent drops in penalty value. This exploration phase is crucial for escaping poor local optima and ensures that the initial solution space is sufficiently sampled. The sudden drops represent successful updates to the global best solution by particles that have discovered more effective ACO parameter combinations. As the iteration progresses toward the midpoint (around iteration 30–50), the frequency of penalty improvements begins to decline. This signifies a transition from global exploration to local exploitation. The particles begin clustering around promising regions of the solution space, refining parameter combinations with more conservative velocity updates. The flattening of the curve after iteration 50 implies that further exploration does not yield significantly better configurations, indicating that a stable local minimum has likely been reached.

From an implementation perspective, this behavior implies that executing more than 50 iterations may lead to diminishing returns in terms of optimization gain. This insight can guide practical decisions in algorithm tuning, such as setting stopping criteria or adjusting inertia weights, to save computation time without sacrificing solution quality. Overall, Figure 7 confirms the effectiveness of PSO in tuning ACO parameters for complex scheduling problems, resulting in improved solution performance through a well-defined convergence process.

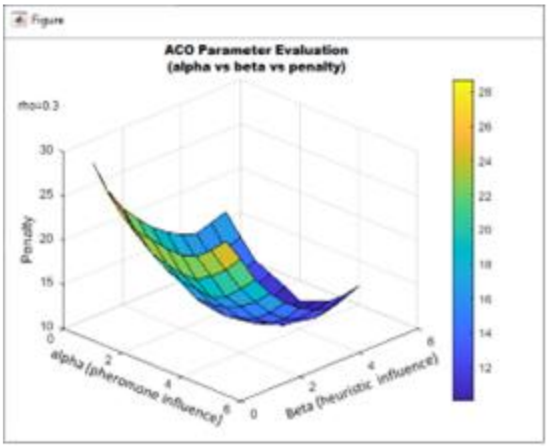

Figure 8. Evaluation of ACO parameter

Figure 8 is a real 3D visualization of ACO evaluation based on alpha and beta parameters, with fixed rho (0.3). This surface illustrates how the combination of alpha-beta values affects ACO performance. The local minimum around (alpha=3, beta = 4) shows the optimal configuration in the region, where the lowest penalty value occurs, indicating the most optimal combination of ACO parameters for the configuration rho = 0.3. This is the sweet spot where the interaction of alpha and beta results in scheduling with minimum overload. It can be seen that there is a correlation between the parameters α and β and the quality of the solution.

Beyond the local minimum, Figure 8 also reveals the sensitivity of the penalty function to variations in alpha and beta values. As either alpha or beta diverges significantly from their optimal values, the penalty increases sharply, as indicated by the gradient of the surface. This pattern implies that an imbalance in the weight given to pheromone trails (alpha) or heuristic visibility (beta) can lead to suboptimal schedules. For instance, high alpha with low beta overly emphasizes historical paths and ignores constructive heuristics, potentially leading to premature convergence. Conversely, high beta with low alpha might result in unstable exploration due to insufficient reinforcement of good paths.

Additionally, the smooth curvature of the surface indicates a continuous and relatively stable response of the penalty to parameter changes, which is ideal for optimization. It suggests that local search methods or gradient-free optimizers like PSO can effectively navigate this surface to find the optimal region. This further validates the use of PSO in parameter tuning, as demonstrated in earlier figures. The lack of sharp discontinuities or rugged terrain implies that the search space is well-behaved and thus suitable for metaheuristic approaches. Moreover, this 3D plot also serves as a visual confirmation of the interaction effects between the parameters. Optimal scheduling performance is not achieved by maximizing or minimizing one parameter alone but through a balanced trade-off. The penalty surface forms a valley-like structure, emphasizing that alpha and beta need to complement each other to guide the artificial ants efficiently. This insight is crucial for practitioners tuning ACO parameters: neither parameter should be optimized in isolation, and automated parameter tuning methods like PSO become essential tools for identifying such balanced configurations.

Figure 9 shows the visualization of the parameter convergence process of the Ant Colony Optimization (ACO) algorithm in the Hybrid ACO–PSO approach for 100 iterations, with three main snapshots: the 5th, 14th, and 100th iterations. The X (α), Y (β), and Z (ρ) axes represent the main parameters of ACO, namely: α (pheromone trail influence), β (heuristic influence), and ρ (pheromone evaporation rate). Blue dots indicate solution particles evaluated by PSO, while red or blue stars indicate the best solution at that iteration. At the 5th iteration, the particles are still widely distributed with the best penalty of 0.0397, indicating that the initial exploration process is still ongoing. Entering the 14th iteration, the particles begin to concentrate, indicating the process of exploitation and solution improvement with a penalty that decreases significantly to 0.0049. At the 100th iteration, all particles have consolidated at one optimal point with a minimum penalty of 0.0001, indicating that the algorithm has achieved optimal convergence with the best ACO parameters for the tested lecture scheduling case. This visualization demonstrates the effectiveness of ACO–PSO hybridization in iteratively refining parameters to achieve high-quality solutions.

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

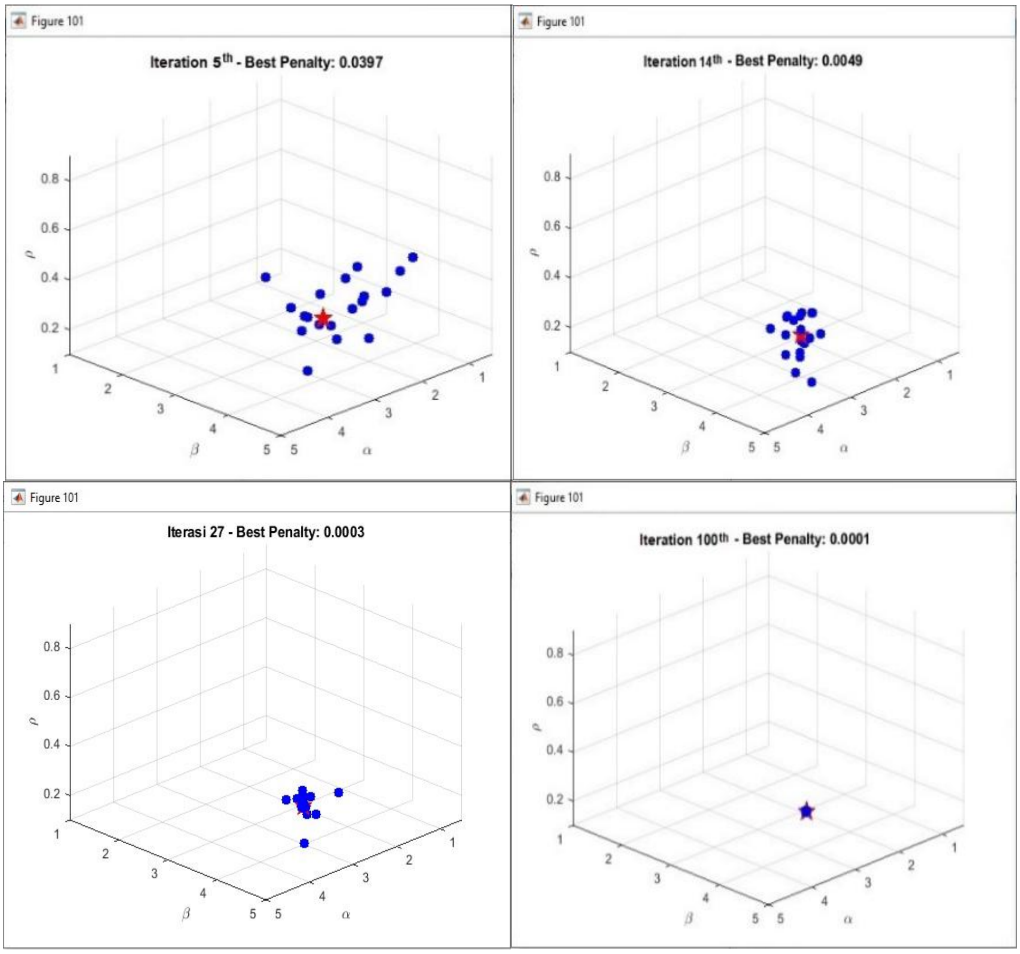

Figure 9. Visualization of the parameter convergence process

In addition to showing the convergence process, Figure 9 also provides insight into the dynamics of exploration and exploitation performed by PSO in the ACO parameter space. In the 5th iteration, particles are spread throughout the search space, including extreme values of α, β, and ρ. This indicates that PSO is still in the global exploration phase, where the main goal is to reach various parameter combinations to map the solution landscape as a whole. This diversity of positions is important to prevent premature convergence and open up opportunities to find optimal configurations in unexpected areas.

Then, in the 14th iteration, the particle distribution begins to form a dense cluster in one region of the parameter space. This indicates that the algorithm has begun to exploit promising solution areas. At this stage, PSO utilizes the best information from individuals and globally (pbest and gbest) to narrow the search to areas with low penalties. This shift from the exploration phase to exploitation is an important transition that indicates efficiency in the solution search.

The next condition shows the convergence of the algorithm parameters at the 27th iteration in the Ant Colony Optimization (ACO) parameter optimization process with the help of Particle Swarm Optimization (PSO). The X, Y, and Z axes each represent the main parameters of ACO, namely: α (the influence of the pheromone trail), β (the influence of heuristic information), and ρ (the pheromone evaporation rate). The distribution of blue dots shows the position of the particles in the parameter space at that iteration, while the blue star, located in the middle of the distribution, indicates the best solution at this iteration with a minimum penalty value of 0.0003. This value indicates that the solution found is very close to optimal, with very small or even no violations of the constraints (hard and soft constraints). It can be seen that all particles have begun to gather in a narrow area in the parameter space. This indicates that the parameter exploration process has begun to shift into an exploitation process, where PSO effectively directs the search to the optimal parameter configuration. The density of particles in this area indicates the stability of the search direction and validation that the parameters found provide very good lecture scheduling solution results. In general, the 27th iteration is a critical point that reflects the success of ACO-PSO hybridization in accelerating the convergence of ACO parameters efficiently and effectively. This visualization also supports the claim that the hybrid approach is able to combine the exploration capabilities of ACO with the global search intelligence of PSO, thus producing optimal parameter configurations in a relatively shorter iteration time.

International Journal of Computer and Information System (IJCIS)
Peer Reviewed – International Journal
Vol : Vol. 06, Issue 03, August 2025
e-ISSN : 2745-9659
https://ijcis.net/index.php/ijcis/index

Finally, at the 100th iteration, all particles have reached the same point, reflecting full convergence. This indicates that the entire population has agreed on a single ACO parameter configuration that is considered optimal. The very low penalty value (0.0001) not only indicates the success of the optimization but also the stability of the solution in the parameter space. This condition shows that the parameters α, β, and ρ found are not only optimal for one solution, but also show high performance consistency against the structure of the problem faced.

Overall, this visualization not only reflects the final result of parameter optimization but also the evolution process of the solution. By observing how the population moves from a wide distribution to a narrow convergence, it can be concluded that the ACO–PSO hybridization is able to carry out the principles of intensification and diversification in a balanced manner. This is very important in combinatorial optimization, such as lecture scheduling, where the quality of the solution is greatly influenced by the right choice of parameters.

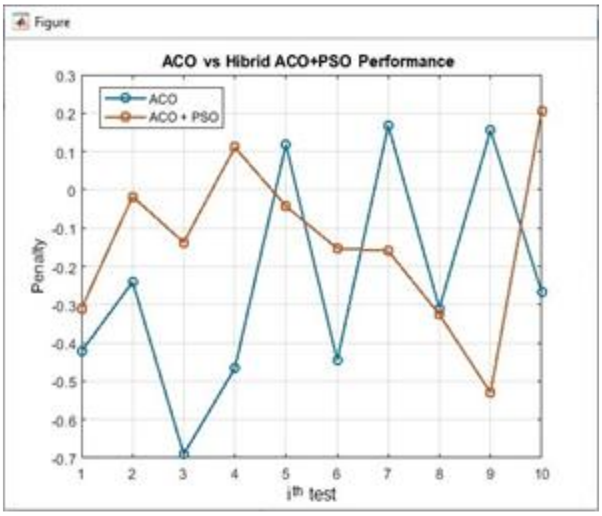

Figure 10. Performance comparison

The graph of Figure 10 compares the performance of the pure ACO algorithm with the hybrid version of ACO+PSO based on the penalty value during 10 trials, where the penalty value indicates the measure of schedule quality (the lower, especially negative, the better the results).

The experiment was conducted 10 times with the same parameters for both approaches. The penalty value is used as an indicator of schedule quality, where the smaller the penalty value indicates the better the solution produced. From the trial result graph, several important findings were obtained: Pure ACO showed significant performance fluctuations. In several trials (for example, the 3rd and 7th trials), pure ACO was able to produce very low penalty values. However, in other trials, its performance decreased quite drastically. This indicates that pure ACO has the potential to produce very good solutions, but is less stable and depends on the initial conditions and parameters used. Meanwhile, hybrid ACO + PSO shows more stable and consistent performance. The resulting penalty value is not as volatile as pure ACO, with a tendency to approach the optimum value on average. This shows that the use of PSO in optimizing ACO parameters helps improve the quality of results in general and reduces performance variability. Thus, it can be analyzed that ACO has the potential to produce very good solutions (very low penalty), but is less consistent. On the other hand, ACO + PSO (hybrid) produces more stable solutions, with smaller fluctuations and better average performance, or in other words, the combination of PSO that optimizes ACO parameters is proven to reduce uncertainty and increase the consistency of algorithm performance.

In addition to comparing the performance stability, Figure 10 also illustrates the differences in adaptive behavior between the pure ACO and ACO+PSO algorithms in responding to the complexity of the problem in each trial. The sharp fluctuations in pure ACO indicate that the solution search often gets stuck at suboptimal parameter values or experiences premature convergence, depending on the initial initialization. This reflects the high sensitivity to variations in initial conditions, which in practice can lead to unreliable results if careful manual parameter tuning is not performed. In contrast, the hybrid ACO+PSO approach shows a more stable curve and is concentrated around a penalty value close to zero. This indicates that the PSO integration successfully adjusts the ACO parameters dynamically for each scenario. This automatic parameter adaptation provides significant advantages in dealing with the case of lecture scheduling that may have different characteristics of conflicts or resource constraints in each trial. Thus, although the best performance of pure ACO occasionally outperforms the hybrid approach, the hybrid approach still has the advantage in terms of reliability. Statistically, the distribution of penalty values of pure ACO is wider than that of hybrid, which means that the variance of the results is higher. This high variability increases the risk of poor results in real implementations. Meanwhile, hybrid results are more centralized, reflecting a more even distribution of performance and indicating that users can expect relatively consistent result quality even when the experiment is repeated. This is important in the context of operational scheduling that demands stable and predictable results.

Table 2 presents a performance comparison between the pure Ant Colony Optimization (ACO) algorithm and the hybrid version of ACO+PSO based on three main aspects, namely performance variation, penalty range, and result stability.

First, in terms of variation or performance variation, pure ACO shows very fluctuating characteristics, with a pattern of results that fluctuate sharply between trials. This indicates that ACO is highly influenced by initial conditions, so that it can produce very good performance in one trial but decline significantly in other trials.

Table 2. Performance comparison

Aspect	ACO	ACO+PSO
Variation	Very volatile, sharp ups and downs	More stable
Value Range	from -0.7 to ~0.3	from -0.4 to ~0.25
Stability	Less stable	More consistent
Best Performance	Sometimes better than hybrid (eg, 3rd, 7th test)	But in general, it's more or less good on average

Furthermore, in terms of value range, pure ACO has a wider penalty value range, namely from -0.7 to around 0.3. This reflects a high level of uncertainty about the quality of the resulting solution. Meanwhile, hybrid ACO+PSO shows a narrower range, namely between -0.4 to around 0.25. Although the minimum value of pure ACO can be better, the narrower range in the hybrid indicates that the quality of the resulting solution tends to be in a good and predictable range. In terms of stability, pure ACO is categorized as less stable, because the sensitivity to initial parameters makes the results consistently unreliable. In contrast, hybrid ACO+PSO is more consistent in producing good solutions. This happens because the PSO process automatically adjusts the ACO parameters, thereby avoiding solutions that are too far from optimal due to poor parameter selection. Finally, in terms of best performance, pure ACO is sometimes able to produce the best performance that even outperforms the hybrid version, as seen in experiments 3 and 7. However, this superiority is sporadic and inconsistent. In contrast, ACO+PSO shows better results on average, even though it is not always the best in every experiment. This makes it a more reliable choice for real implementations, because the results are not only good but also consistent. Overall, the interpretation of Table 2 shows that although pure ACO has the potential for high performance, the hybrid ACO+PSO approach excels in terms of stability, consistency, and average overall performance, making it a more robust solution in complex scheduling scenarios.

Finally, it can be asserted that the role of PSO as a parameter optimizer in ACO not only improves the average performance, but also significantly improves the stability and predictability of the algorithm. Thus, this hybrid approach is more feasible to be used in real implementations for complex scheduling systems such as lecture scheduling, where the consistency and quality of the solution are crucial.

VI. CONCLUSION

This study aims to evaluate and compare the performance of the pure Ant Colony Optimization (ACO) algorithm with the hybrid ACO approach combined with Particle Swarm Optimization (PSO) in solving class scheduling problems. The hybrid ACO+PSO approach is superior in terms of consistency and average performance, although pure ACO still has the advantage in its excellent solution exploration capabilities under certain conditions. The combination of these two algorithms can be an effective approach in solving class scheduling optimization problems more reliably.

Furthermore, the visualization of the parameter convergence process clearly demonstrates how the hybrid ACO+PSO method successfully refines ACO parameters over successive iterations. The gradual clustering of parameter combinations, as depicted in the convergence plots, indicates an efficient balance between exploration and exploitation during the optimization process. This convergence leads to a significant reduction in penalty values, reflecting improved scheduling quality and confirming the capability of the hybrid method to reach near-optimal parameter settings systematically.

In addition, experimental comparisons between the two methods across multiple trials reveal that the hybrid ACO+PSO consistently produces more stable results with lower variation. While the pure ACO algorithm can occasionally outperform the hybrid in isolated cases, its high performance volatility makes it less reliable for practical applications. The statistical consistency and reduced fluctuation of the hybrid approach suggest that PSO plays a crucial role in enhancing ACO's robustness, making it better suited for real-world scheduling scenarios where solution reliability is a key requirement.

REFERENCES

[1] O.S. Adewale, I.E. Onwuka, I.E. Kingsley, "A Tabu Search-based University Lectures Timetable Scheduling Model," *International Journal of Computer Applications*, vol. 181, no. 9, pp. 16-23, 2018. DOI=10.5120/ijca2018917599

[2] M. Yazdani, B. Naderi, E. Zeinali, "Algorithms for University Course Scheduling Problems," *Tehnički vjesnik,* vol. 24, no. 2, pp. 241-247, 2017.

[3] S. Yarat, S. Senan, Z. Orman, "A Comparative Study on PSO with Other Metaheuristic Methods," *Mercangöz, B.A. (eds) Applying Particle Swarm Optimization. International Series in Operations Research & Management Science*, Springer, Cham, vol. 306, 2021. https://doi.org/10.1007/978-3-030-70281-6_4

[4] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, A. Baroutaji, "Hybrid metaheuristic algorithms: a recent comprehensive review with bibliometric analysis," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 14, no. 6, pp. 7022-7035, 2024. DOI: 10.11591/ijece.v14i6.

[5] J. Lu, W. Hu, Y. Wang, L. Li, P. Ke, K. Zhang, "A Hybrid Algorithm Based on Particle Swarm Optimization and Ant Colony Optimization Algorithm," *Qiu, M. (eds) Smart Computing and*

Communication. Lecture Notes in Computer Science, Springer, Cham, vol 10135, 2017. https://doi.org/10.1007/978-3-319-52015-5_3

[6] N. Bhatia, P. Chauhan, H. Yadav, "Applications of Hybrid Particle Swarm Optimization Algorithm: A Survey," *Lecture Notes in Networks and Systems*, Springer, Singapore, vol 166, 2021. https://doi.org/10.1007/978-981-15-9689-6_32

[7] R. Dietze, M. Berger, "A Hybrid Particle Swarm Optimization and Hill Climbing Algorithm for Task Scheduling on Heterogeneous Multicore Clusters," *Lecture Notes in Networks and Systems*, Springer, Cham., vol 1346, 2025. https://doi.org/10.1007/978-3-031-87647-9_1

[8] B. Shuang, J. Chen, and Z. Li, "Study on hybrid PS-ACO algorithm," *Appl Intell,* vol. **34**, pp. 64–73, 2011. https://doi.org/10.1007/s10489-009-0179-6

[9] S. Akter, M.H. Khan, L. Nishat, F. Alam, A.W. Reza, M.S. Arefin, "A Hybrid Approach for Improving Task Scheduling Algorithm in the Cloud," *Intelligent Computing and Optimization, Lecture Notes in Networks and Systems*, Springer, Cham, vol 854, 2023. https://doi.org/10.1007/978-3-031-50151-7_18

[10] L. Jie, "Optimizing Resource Utilization and Improving Performance in Cloud Computing Through PSO-Based Scheduling and ACO-Based Load Balancing," *J. Inst. Eng. India Ser. B* (2024). https://doi.org/10.1007/s40031-024-01139-3

[11] F. Yunita, Pranowo, and A.J. Santoso, "Hybrid model of particle swarm and ant colony optimization in lecture schedule preparation," *AIP Conference Proceedings*, vol. 1977, no. 020039, 2018. https://doi.org/10.1063/1.5042895

[12] S. Kaliappan, V. Paranthaman, M. D. R. Kamal, S. Avv, and M. Muthukannan, "A Novel Approach of Particle Swarm and Ant Colony Optimization for Task Scheduling in Cloud," *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence),* Noida, India, pp. 272-278, 2024. DOI: 10.1109/Confluence60223.2024.10463398.

[13] J. Lu, A. Teng, J. Zha, L. Shen and Z. Wang, "Cloud Computing Task Scheduling Strategy Based on Improved Ant Colony Optimization (ACO) Algorithm," *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China, pp. 1368-1373, 2024. DOI: 10.1109/ICPICS62053.2024.10796062.

[14] C. Chandrashekar, P. Krishnadoss, V. K Poornachary, B. Ananthakrishnan, and K. Rangasamy, "HWACOA Scheduler: Hybrid Weighted Ant Colony Optimization Algorithm for Task Scheduling in Cloud Computing," *Applied Sciences*, vol. 13, no. 6, 3433, 2023. https://doi.org/10.3390/app13063433

[15] N. Jananeeswari, S. Jayakumar, and M. Nagamani, "Multi-Objective for a Partial Flexible Open Shop Scheduling Problem using Hybrid Based Particle Swarm Algorithm and Ant Colony Optimization," *International Journal of Mathematics Trends and Technology-IJMTT* 44, vol. 44, no. 2, 2017. DOI :10.14445/22315373/IJMTT-V44P520

[16] T. R. Mahesh, D. Santhakumar, A. Balajee, H. S. Shreenidhi, V. V. Kumar, and J. R. Annand, "Hybrid Ant Lion Mutated Ant Colony Optimizer Technique with Particle Swarm Optimization for Leukemia Prediction Using Microarray Gene Data," *IEEE Access*, vol. 12, pp. 10910-10919, 2024. DOI: 10.1109/ACCESS.2024.3351871.

[17] E. Koyuncu, R. Erol, "PSO-based approach for scheduling NPD projects including overlapping process," *Computers & Industrial Engineering*, vol. 85, pp. 316-327, 2015.

[18] G.S. Rao, C.V.P. Krishna, K.R. Rao, "Multi-Objective Particle Swarm Optimization for Software Cost Estimation," *Advances in Intelligent Systems and Computing*, Springer, Cham, vol 248, 2014. https://doi.org/10.1007/978-3-319-03107-1_15

[19] M.N. A. Wahab, S.N. Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS One*, vol. 10, no. 5: e0122827, 2015. DOI: 10.1371/journal.pone.0122827.

[20] C.C. Bolton, V. Parada, "Automatic Combination of Operators in a Genetic Algorithm to Solve the Traveling Salesman Problem," *PLoS One*, vol.10, no. 9: e0137724, 2015. DOI: 10.1371/journal.pone.0137724.

[21] Y. Ge, B. Xu, "Dynamic Staffing and Rescheduling in Software Project Management: A Hybrid Approach," *PLoS One*, vol. 11, no. 6: e0157104, 2016. DOI: 10.1371/journal.pone.0157104.