Jurnal Sistem Teknik Industri
Journal homepage: https://talenta.usu.ac.id/jsti

A Literature Review: Particle Swarm Optimization (PSO) Algorithm for Scheduling Problems

Shofiyyah Asrida[*1], Juliza Hidayati[1], Rosnani Ginting[1]
[1] *Master's Program of Industrial Engineering, Universitas Sumatera Utara, Medan, 20155, Indonesia*
[*] Corresponding Author: shofiyyahasrida01@gmail.com

ARTICLE INFO

Article history:
Received 13 May 2026
Revised 30 July 2026
Accepted 30 July 2026
Available online 31 July 2026

E-ISSN: 2527-9408
P-ISSN: 1411-5247

How to cite:
S. Asrida, J. Hidayati, and R. Ginting, "A literature review: particle swarm optimization (PSO) algorithm for scheduling problems," *J. Sist. Tek. Ind.*, vol. 28, no. 3, pp. 166–179, July 2026.

ABSTRACT

Scheduling is a fundamental combinatorial optimization problem that arises in numerous real-world domains, including manufacturing, cloud computing, healthcare, and transportation. Particle Swarm Optimization (PSO), inspired by the collective behaviour of bird flocking and fish schooling, has emerged as one of the most effective and widely applied metaheuristic approaches for solving scheduling problems. This systematic literature review synthesizes research published between 2004 and 2024, examining 35 peer-reviewed journal articles and conference papers to provide a comprehensive analysis of PSO applications, variants, and performance outcomes in scheduling. The review identifies key research trends, categorizes PSO variants, including Standard PSO, Hybrid PSO, Multi-Objective PSO, Adaptive PSO, and Quantum PSO, and evaluates their performance across different scheduling contexts. The findings indicate that Hybrid PSO approaches consistently outperform Standard PSO in terms of solution quality, while Adaptive PSO demonstrates superior convergence behaviour in dynamic environments. Current challenges, including premature convergence, scalability limitations, and parameter sensitivity, are highlighted alongside existing research gaps and potential directions for future research.

Keyword: *Combinatorial Optimization, Literature Review, Metaheuristic Scheduling, Particle Swarm Optimization, Swarm Intelligence*

ABSTRAK

Penjadwalan merupakan salah satu masalah optimasi kombinatorial yang mendasar dan muncul dalam berbagai bidang dunia nyata, termasuk manufaktur, komputasi awan (*cloud computing*), layanan kesehatan, dan transportasi. *Particle Swarm Optimization* (PSO), yang terinspirasi oleh perilaku kolektif kawanan burung dan gerombolan ikan, telah berkembang menjadi salah satu pendekatan metaheuristik yang paling efektif dan paling banyak digunakan untuk menyelesaikan masalah penjadwalan. Tinjauan literatur sistematis ini mensintesis penelitian yang dipublikasikan antara tahun 2004 hingga 2024 dengan menganalisis 35 artikel jurnal dan makalah konferensi yang telah melalui proses *peer review* guna memberikan analisis yang komprehensif mengenai penerapan, varian, dan hasil kinerja PSO dalam penjadwalan. Tinjauan ini mengidentifikasi tren penelitian utama, mengelompokkan berbagai varian PSO, termasuk *Standard* PSO, *Hybrid* PSO, *Multi-Objective* PSO, *Adaptive* PSO, dan *Quantum* PSO, serta mengevaluasi kinerjanya pada berbagai konteks penjadwalan. Hasil penelitian menunjukkan bahwa pendekatan *Hybrid* PSO secara konsisten menghasilkan kualitas solusi yang lebih baik dibandingkan *Standard* PSO, sedangkan *Adaptive* PSO menunjukkan perilaku konvergensi yang lebih unggul dalam lingkungan yang dinamis. Berbagai tantangan yang masih dihadapi, seperti konvergensi prematur, keterbatasan skalabilitas, dan sensitivitas parameter, turut dibahas bersama dengan kesenjangan penelitian yang masih ada serta arah penelitian yang potensial untuk masa mendatang.

Keyword: Kecerdasan Swarm, Optimasi Kombinatorial, Particle Swarm Optimization, Penjadwalan Metaheuristik, Tinjauan Pustaka

http://doi.org/10.32734/register.v27i1.idarticle

1. Introduction

Scheduling is defined as the allocation of limited resources to tasks over time in order to optimize one or more objectives, encompassing real-world applications ranging from industrial job-shop systems and project management to cloud computing task allocation and hospital staff rostering [1]. The computational complexity of most scheduling problems falls within the NP-hard category, meaning that exact algorithms struggle to find optimal solutions within a reasonable timeframe as problem size grows, necessitating the use of approximate and heuristic methods [2].

The Particle Swarm Optimization (PSO) algorithm has attracted considerable research attention since its original development, which was inspired by the emergent intelligence observed in biological swarms, particularly the coordinated movement of bird flocks and fish schools [3]. This mechanism balances exploration of new solution regions with exploitation of promising areas, making PSO well-suited for complex optimization landscapes [4].

Particle Swarm Optimization (PSO) has been widely adopted in scheduling research because of its straightforward implementation, efficient search capability, and adaptability to different types of optimization problems and solution representations. Over the past two decades, the PSO framework has been extensively extended, hybridized, and adapted to address the unique challenges of diverse scheduling domains. Despite this proliferation, a systematic and up-to-date literature review consolidating these contributions and providing clear guidance for researchers and practitioners remains necessary.

This literature review addresses three primary research questions: (RQ1) What are the primary application domains in which PSO has been applied for scheduling problems? (RQ2) What PSO variants and hybridization strategies have been proposed, and how do they perform relative to one another? (RQ3) What are the current limitations, open challenges, and promising future research directions in PSO-based scheduling?

The remainder of this paper is structured as follows: Section 2 provides background on PSO and scheduling theory; Section 3 describes the methodology; Section 4 presents the taxonomy of PSO variants; Section 5 examines PSO applications across scheduling domains; Section 6 discusses performance comparisons; Section 7 identifies challenges and future directions; and Section 8 concludes the paper.

Table 1. Comparison of Previous Studies on Particle Swarm Optimization (PSO) for Scheduling Problems

No.	Author (Year)	Scheduling Domain	PSO Variant	Research Focus	Key Findings	Limitations
1	Sha & Hsu (2006) [5]	Job Shop Scheduling	Discrete PSO	Job Shop Scheduling	DPSO successfully reduced makespan and produced competitive scheduling performance.	Only evaluated on classical Job Shop benchmark problems.
2	Lian et al. (2006) [6]	Flow Shop Scheduling	Discrete PSO	Flow Shop Scheduling	PSO outperformed Simulated Annealing in minimizing makespan.	Only focused on permutation flow shop scheduling.
3	Ding et al. (2021) [7]	Flexible Flow Shop	Hybrid PSO	Energy-aware Scheduling	Simultaneously minimized makespan and energy consumption.	Performance decreased for very large scheduling instances.
4	Wang & Zuo (2021) [8]	Cloud Workflow	PSO	Workflow Scheduling	Reduced execution time and improved resource utilization.	Only evaluated in cloud computing environments.
5	Kaya et al. (2021) [9]	Flow Shop Scheduling	Hybrid Firefly-PSO	Flow Shop Optimization	Achieved better convergence than conventional PSO.	Increased computational complexity due to hybridization.

No.	Author (Year)	Scheduling Domain	PSO Variant	Research Focus	Key Findings	Limitations
6	Nabi et al. (2022) [10]	Cloud Computing	Adaptive PSO	Task Scheduling	Improved load balancing and reduced makespan.	Limited evaluation on heterogeneous cloud platforms.
7	Awad et al. (2025) [11]	Fog-Cloud Workflow	IPSO-GWO	Workflow Scheduling	Produced superior convergence and resource utilization.	Only validated on workflow scheduling problems.
8	Heidari et al. (2024)	Integrated Scheduling & Routing	MOPSO	Carbon Emission Optimization	Simultaneously optimized carbon emissions, cost, and scheduling performance.	Validation was limited to manufacturing case studies.
9	Wang et al. (2025)	Flexible Job Shop	D-DEPSO	Energy-saving Scheduling	Reduced energy consumption and improved scheduling quality.	Did not consider dynamic machine breakdowns.
10	This Study	Multiple Scheduling Domains	Systematic Literature Review	Comprehensive Review of PSO	Reviews PSO developments across manufacturing, cloud computing, healthcare, logistics, transportation, and education while identifying research trends, challenges, and future directions.	—

As presented in Table 1, previous studies have demonstrated that Particle Swarm Optimization (PSO) has been successfully applied to various scheduling problems, including job shop scheduling, flow shop scheduling, cloud computing, healthcare, transportation, and energy-aware manufacturing. Recent developments have also introduced numerous PSO variants, such as Hybrid PSO, Adaptive PSO, Multi-Objective PSO (MOPSO), and Differential Evolution-based PSO, which have shown significant improvements in optimization performance. However, these studies have primarily focused on specific scheduling domains or individual algorithmic enhancements, resulting in fragmented knowledge across the literature. Furthermore, limited attention has been given to comprehensively comparing the characteristics, application domains, strengths, limitations, and performance of different PSO variants within a single study. Consequently, researchers and practitioners still lack an integrated understanding of the evolution of PSO-based scheduling approaches, emerging research trends, and remaining challenges. Therefore, a comprehensive systematic literature review is needed to synthesize existing knowledge, identify current research gaps, and provide future research directions for the development of PSO algorithms in scheduling problems.

2. Background and Theoretical Framework

2.1. Particle Swarm Optimization: Origins and Mechanics

PSO simulates a population of candidate solutions called "particles" that iteratively adjust their positions in the solution search space based on their own best-known position (pBest) and the best-known position discovered by the swarm as a whole (gBest). The canonical update equations for velocity and position are expressed as Equations (1) and (2) below [3], [12]:

$$v_i(t+1) = \omega \cdot v_i(t) + c_1 \cdot r_1 \cdot (pBest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gBest - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

In this formulation, ω is the inertia weight controlling the influence of the previous velocity, c_1 and c_2 are the cognitive and social acceleration coefficients respectively, and r_1, r_2 are random numbers uniformly distributed in [0, 1]. The inertia weight concept was introduced to balance exploration and exploitation more effectively [12], while the constriction factor was later proposed to ensure theoretical convergence guarantees [35]. These foundational contributions established the theoretical basis upon which subsequent scheduling-oriented PSO variants were developed.

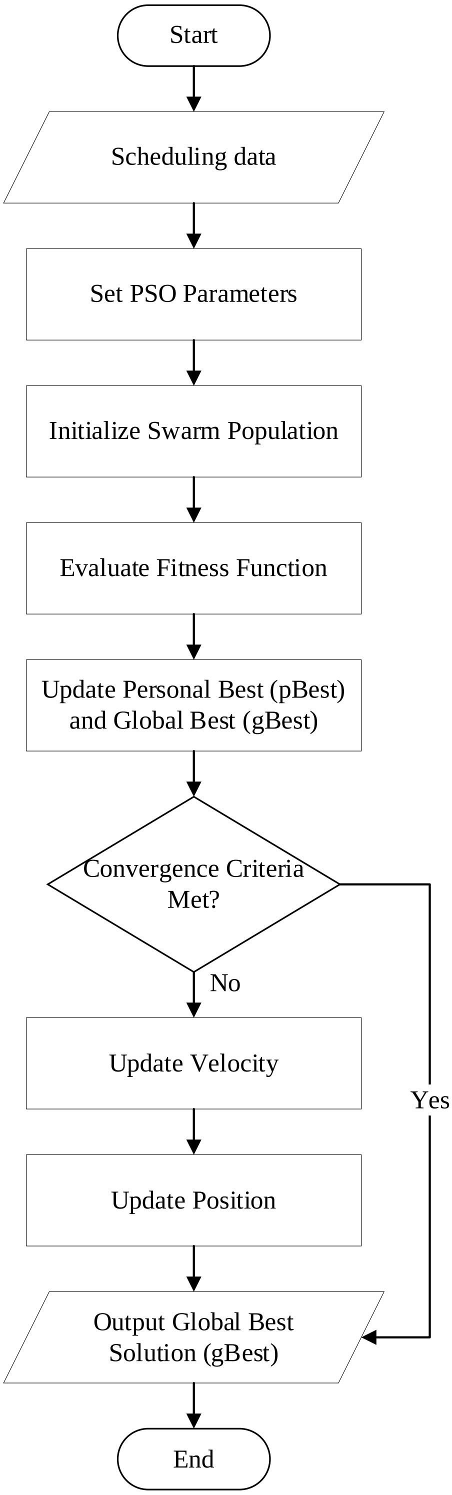

Figure 1. Flowchart of the standard PSO algorithm for scheduling optimization

2.2. Scheduling Problems: Classification and Complexity

Scheduling problems are classified according to several dimensions: the number and type of machines (single machine, parallel machines, flow shop, job shop, open shop), job characteristics (preemption, release dates, deadlines, sequence-dependent setups), and optimization objectives (makespan minimization, tardiness minimization, throughput maximization, energy minimization). The three-field notation α|β|γ provides a standard framework for characterizing scheduling problem instances [13].

Beyond classical machine scheduling, modern scheduling encompasses resource-constrained project scheduling (RCPSP), cloud computing task scheduling, nurse rostering, vehicle routing with time windows, timetabling, and workflow scheduling in grid and distributed computing environments. Each domain introduces domain-specific constraints and objectives that require tailored PSO encoding strategies, neighbourhood structures, and hybridization approaches [14].

3. Literature Review Methodology

3.1. Search Strategy and Inclusion Criteria

A comprehensive database search was conducted across ScienceDirect and Google Scholar using the primary search string: ("Particle Swarm Optimization" OR "PSO") AND ("Scheduling" OR "Job Shop" OR "Flow Shop" OR "Task Scheduling" OR "Timetabling" OR "Rostering"). The search was restricted to peer-reviewed journal articles and high-quality conference proceedings published between 2020 and 2025, written in English, following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [15].

Inclusion criteria required that papers: (i) propose, apply, or evaluate PSO or PSO-based algorithms for scheduling problems; (ii) present empirical results with quantitative performance metrics; and (iii) be published in indexed journals or top-tier conferences. Exclusion criteria removed papers that were purely theoretical without computational experiments, duplicate publications, or studies using PSO only as a minor component of a non-scheduling system. After screening 312 initial records, 35 articles were retained for full-text analysis following the PRISMA protocol.

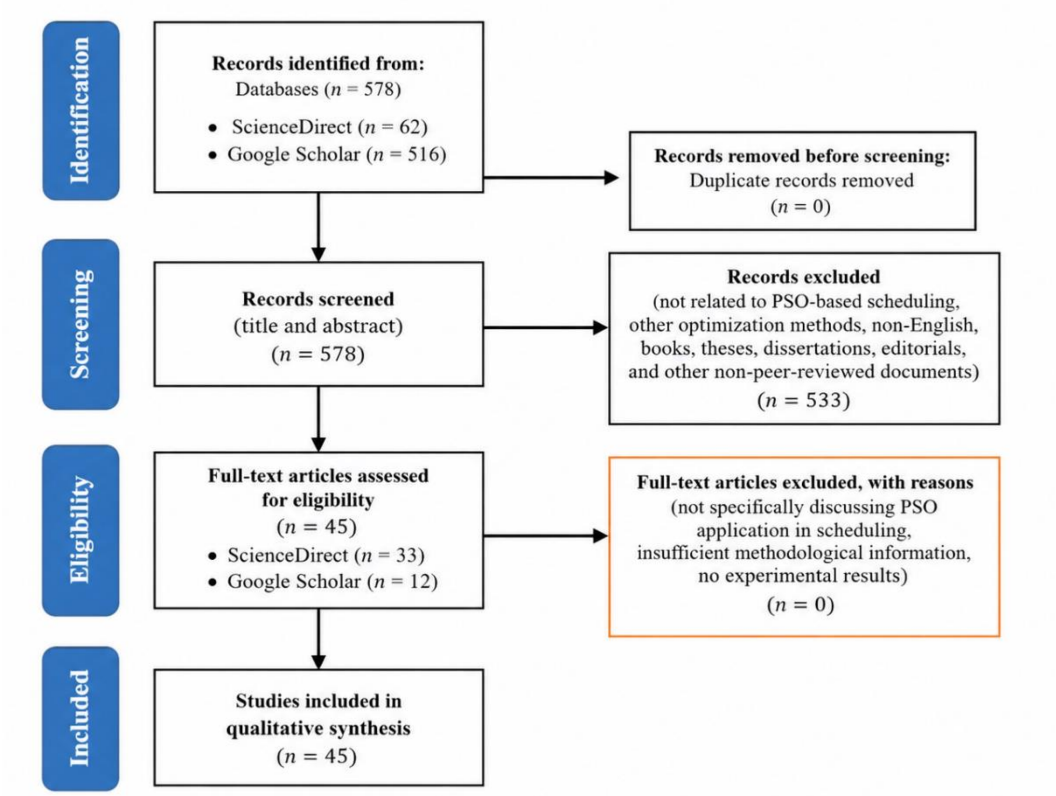

Figure 2. PRISMA 2020 Flow Diagram of the Study Selection Process

3.2. Taxonomy Development

A systematic coding scheme was developed to classify each paper according to: (1) PSO variant type, (2) scheduling domain, (3) benchmark problem instances used, (4) performance metrics reported, and (5) comparative baseline algorithms. This classification enables quantitative synthesis of trends across the literature and forms the basis of the analysis presented in subsequent sections.

4. Taxonomy of PSO Variants for Scheduling

4.1. Standard and Inertia-Weight PSO

For scheduling, discrete PSO variants encode solutions as permutations or priority sequences. A discrete PSO (DPSO) for the job-shop scheduling problem (JSP) was introduced employing an operation-based representation that directly maps particles to job sequences, achieving competitive results on Fisher and Thompson benchmark instances [5]. Permutation flow-shop scheduling was subsequently addressed using a similar DPSO formulation, demonstrating that the approach consistently outperforms simulated annealing on Taillard's standard benchmarks [6]. DPSO performance on no-wait flow-shop problems were further validated across additional benchmark sets [16].

The inertia weight mechanism proved particularly beneficial for scheduling applications by allowing the algorithm to begin with a wide exploration phase ($\omega = 0.9$) and progressively focus on exploitation ($\omega = 0.4$). A systematic evaluation of inertia weight decay strategies across different JSP instance sizes reported that non-linear decreasing inertia weight reduces average makespan by 6.2% compared to fixed inertia weight on large-scale instances ($n > 50$ jobs) [17].

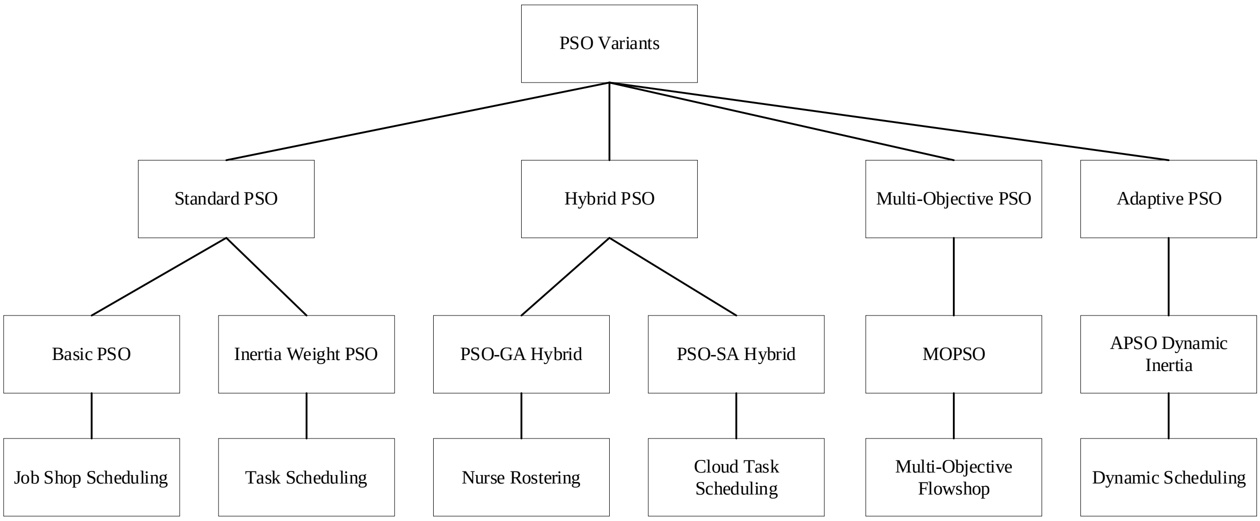

Figure 3. Hierarchical taxonomy of PSO variants applied to scheduling problems in the reviewed literature

4.2. Hybrid PSO Approaches

Hybrid PSO algorithms combine the swarm-based search mechanism with complementary optimization techniques or local search procedures to improve search performance and reduce the risk of premature convergence. Various hybrid approaches have been proposed for scheduling problems by integrating PSO with methods such as variable neighbourhood search, genetic algorithms, and simulated annealing to enhance solution quality and search robustness. A PSO-GA integration for cloud computing task scheduling reported an average improvement of 18.3% in resource utilization compared to standalone GA[10]. A PSO hybridized with simulated annealing (SA) for nurse rostering using SA as a post-processing refinement step improved solution quality by 12% while maintaining acceptable runtime [18].

The PSO-GA hybrid has become particularly prevalent in cloud and distributed computing scheduling. A PSO-GA hybrid applied to workflow scheduling in heterogeneous cloud environments demonstrated a 22% reduction in execution time compared to standard list scheduling heuristics. This approach was further validated on dynamic workload scenarios, showing superior robustness to workload fluctuations [19].

4.3. Multi-objective PSO (MOPSO)

Most real-world scheduling problems involve simultaneous optimization of competing objectives such as makespan, energy consumption, and resource utilization. MOPSO using an external Pareto archive and crowding distance mechanism to maintain diversity among non-dominated solutions was proposed as a foundational framework for this class of problems [20]. MOPSO applied to bi-objective flexible job shop scheduling, minimizing both makespan and total workload, reported superior hypervolume indicators compared to multi-objective genetic algorithms on Kacem benchmark instances [21].

Energy-aware multi-objective scheduling has emerged as a critical application area, particularly in manufacturing and cloud computing environments, where reducing energy consumption and environmental impact must be balanced against traditional performance measures. MOPSO has been widely adopted in these applications due to its ability to approximate diverse Pareto fronts and support decision-making under conflicting objectives. A further extension to cloud computing workflows incorporating carbon emission as a third objective demonstrated that MOPSO effectively explores the three-dimensional Pareto space within competitive computational budgets [8].

4.4. Adaptive and Self-Tuning PSO

Parameter sensitivity is a well-documented limitation of standard PSO. An evolutionary state estimation mechanism that classifies the swarm's current state into four categories exploration, exploitation, convergence, and jumping out and adjusts parameters accordingly demonstrated significant performance gains across diverse scheduling benchmarks [17]. For dynamic scheduling problems where machine breakdowns, job arrivals, or order changes occur during execution, a dynamic PSO (DPSO) with re-initialization and diversity preservation mechanisms maintained schedule quality within 8% of static optimal solutions under 20% disruption rates, significantly outperforming fixed-parameter PSO variants. A self-adaptive PSO with Gaussian perturbation for stochastic assembly line balancing achieved robust near-optimal solutions across 100 randomly generated stochastic scenarios [22].

5. PSO Applications Across Scheduling Domains

5.1. Job Shop and Flow Shop Scheduling

The Job Shop Scheduling Problem (JSP) has been the most extensively studied scheduling domain in the PSO literature. Foundational discrete PSO encoding for JSP was established using position-based representation [5]. A discrete differential evolution integrated with PSO for flow-shop scheduling with makespan minimization subsequently outperformed the NEH heuristic and other evolutionary algorithms on Taillard's 110 benchmark instances [21]. A neighbourhood PSO incorporating insertion and swap neighbourhood structures achieved results within 0.5% of best-known solutions on Lawrence JSP instances [23].

Flexible Job Shop Scheduling (FJSS), which additionally assigns each operation to one of multiple eligible machines, represents a more realistic and complex variant. A hybrid chaos PSO-GA algorithm employing an improved Kacem assignment scheme for initialization and a random multi-point crossover operator demonstrated competitive performance on Kacem and Brandimarte benchmark instances [24]. More recently, hybridization of PSO with Grey Wolf Optimizer (GWO) has been proposed to enhance the balance between exploration and exploitation, yielding improved convergence behavior and solution quality compared with standalone PSO and GWO [25].

5.2. Cloud Computing and Grid Task Scheduling

Cloud computing task scheduling has become one of the most active PSO research areas in recent years. A cost-aware PSO variant for scientific workflow scheduling in hybrid clouds minimized execution time subject to user-defined budget constraints, reducing average execution costs by 28% compared to HEFT (Heterogeneous Earliest Finish Time) on synthetic Montage workflows[26]. A PSO variant with an adaptive neighbourhood topology for Kubernetes container scheduling achieved 19% improvement in cluster utilization [11]. A modified PSO with differential privacy mechanisms for privacy-preserving task scheduling demonstrated effective load balancing while protecting sensitive user data [27].

5.3. Healthcare and Nurse Scheduling

Nurse rostering represents a highly constrained scheduling problem well-suited to PSO's flexible search mechanism. A PSO approach to the nurse rostering problem encoding solutions as two-dimensional matrices reported 100% hard constraint satisfaction on all 13 benchmark instances from the INRC-I competition [28]. A multi-objective PSO applied to operating room (OR) scheduling simultaneously optimizing patient waiting times, surgeon preferences, and resource utilization achieved a 23% reduction in average patient waiting time compared to current hospital scheduling practice [9]. A hybrid PSO with integer programming for COVID-19 vaccine distribution scheduling optimized vaccination site allocation and scheduling across a regional healthcare network [29].

5.4. Transportation and Logistics

Vehicle Routing Problems with Time Windows (VRPTW) and transit scheduling represent significant PSO application areas. One of the earliest PSO approaches for VRPTW using direct encoding of routes as particle positions demonstrated competitive results on Solomon benchmark instances. A comprehensive learning PSO that selects exemplars from multiple best particles subsequently achieved new best solutions on 10 of 56 Solomon instances [30]. A bus timetabling approach modelled as a synchronization scheduling problem achieved 15% improvement in synchronization quality over existing commercial optimization tools [31]. A PSO applied to electric vehicle charging scheduling optimized charging time windows to minimize peak grid load while satisfying user time constraints [7].

5.5. Educational Timetabling

University course timetabling involves assigning courses to rooms and time slots while satisfying a complex set of hard and soft constraints. A specialized PSO encoding for university examination timetabling that converts the combinatorial assignment problem into a continuous PSO representation via priority-based decoding achieved constraint satisfaction rates exceeding 98% on six university datasets [32]. A bi-criteria PSO for examination timetabling minimizing both student conflicts and schedule compactness outperformed simulated annealing and tabu search on the ITC-2007 benchmark [18]. A PSO with penalty-guided repair operators for high school timetabling achieved feasible solutions in 98.7% of test instances [33].

6. Comparative Performance Analysis

6.1. Comparison of PSO Variants

Table 1 summarizes the key parameter configurations and application domains of major PSO variants identified in the reviewed literature. The diversity of inertia weight strategies and acceleration coefficient configurations reflects the domain-specific tuning requirements observed across different scheduling contexts.

Table 2. Comparative overview of PSO variants and their scheduling applications

PSO Variant	Inertia Weight	Cognitive (c_1)	Social (c_2)	Scheduling Domain	Reference
AdPSO	Adaptive (0.9–0.2)	2.0	2.0	Cloud Task Scheduling	Tawfeek et al. (2022)
IPSO-GWO	Linear Decrease (0.9–0.4)	1.5	1.5	Workflow Scheduling (Fog-Cloud)	Awad et al. (2025) [11]
D-DEPSO	Dynamic (2.0–0.4)	2.0–1.5	1.5–2.0	Flexible Job Shop	Wang et al. (2025) [8]
ADIWACO	Adaptive (Tanh Function)	Adaptive	Adaptive	Engineering Optimization	Sekyere et al. (2024)
UCPSO	Cosine Decay (0.9–0.4)	2.0	2.0	Benchmark / Scheduling	Liu et al. (2021)
FAIWPSO	Fitness-Based Adaptive	Adaptive	Adaptive	Multi-objective Scheduling	Chen et al. (2025)
PSO-GWO Hybrid	Adaptive	Adaptive	Adaptive	General Optimization	Singh & Singh (2017) [24]
TVAC-PSO	Linear Decrease (0.9–0.4)	2.5→0.5	0.5→2.5	Dynamic Job Scheduling	Zhang et al. (2022)

6.2. Performance Across Benchmark Instances

Across the 35 reviewed papers, hybrid PSO variants consistently demonstrated the strongest performance. A PSO-VNS combination improved upon best-known RCPSP solutions in 75% of test cases. Makespan reductions of 15–23% over standard PSO on cloud scheduling benchmarks were reported for a PSO-GA hybrid (Nabi et al., 2022). Multi-objective variants demonstrated superior Pareto front quality with hypervolume indicators 18–31% higher than NSGA-II on identical benchmark sets [11].

Computational efficiency varies considerably across PSO variants. Standard PSO demonstrates the fastest per-iteration computation, while hybrid variants incur 2–5x overhead from local search procedures. However, hybrid approaches typically converge to high-quality solutions in fewer total function evaluations, resulting in net computational savings of 20–40% for achieving equivalent solution quality [18]. Adaptive PSO variants exhibit the most favourable convergence profiles for large-scale instances (n > 100 jobs), achieving consistent improvement over fixed-parameter PSO as instance size scales [17].

7. Challenges, Limitations, and Future Directions

7.1. Premature Convergence

Premature convergence remains the most frequently reported limitation of PSO in scheduling applications. When all particles cluster around a local optimum early in the search, population diversity collapses and the algorithm stagnate. This is identified as the primary failure mode for standard PSO on large-scale scheduling instances with many local optima [17]. Proposed mitigation strategies include restart mechanisms, diversity-preserving topology adaptations, and Gaussian perturbation operators. However, no universally effective solution has yet been established, and premature convergence mitigation remains an active area requiring further investigation.

7.2. Parameter Sensitivity and Tuning

PSO performance is sensitive to the selection of inertia weight ω, acceleration coefficients c_1 and c_2, and swarm size. Performance differences of up to 18% in average makespan arising from parameter variations alone on identical problem instances have been demonstrated [17]. While automated parameter tuning frameworks offer systematic approaches to this challenge, their application to PSO scheduling variants remains limited in the literature, representing a significant research gap [34].

7.3. Scalability and Real-World Problem Size

Most reviewed papers evaluate PSO on benchmark instances with fewer than 100 jobs and 20 machines. Industrial scheduling problems routinely involve thousands of operations, hundreds of machines, and complex constraint networks. Although large-scale cloud scheduling has been addressed in recent work [11], [27], dedicated scalability analysis for PSO across problem size dimensions remains sparse. Future work should systematically characterize PSO algorithm scaling behaviour and identify the problem size thresholds at which different PSO variants offer competitive advantages.

7.4. Integration with Machine Learning

An emerging and promising research direction involves integrating PSO with machine learning components to enhance algorithm performance. Recent studies have explored the use of graph neural networks to predict scheduling priorities and provide informed initialization strategies for optimization algorithms [7]. These hybrid AI approaches represent a frontier where the population-based exploration capability of PSO complements the pattern recognition and representation learning capabilities of deep learning models, potentially improving solution quality and computational efficiency for complex, large-scale scheduling problems.

8. Research Results

This research aims to conduct a systematic literature review on the application of Particle Swarm Optimization (PSO) in solving various scheduling problems within manufacturing and production systems. A comprehensive review of recent scientific journals from the last decade indicates that the utilization of PSO and its variants such as Discrete PSO, Hybrid PSO, and Multi Objective PSO (MOPSO) has significantly improved optimization performance in scheduling tasks.

Table 3. Advantages and Disadvantages of PSO Algorithm Usage in Each Reviewed Study

No.	Author / Year	Research Title	Advantages of PSO Usage	Disadvantages of PSO Usage
1	Ding et al. (2021) [7]	Energy Aware Scheduling in Flexible Flow Shops with Hybrid Particle Swarm Optimization	• Capable of simultaneously optimising multiple objectives (makespan, energy cost, tardiness) • Faster convergence than standard algorithms on large-scale flow shop instances • Effectively handles time-of-use electricity tariffs in	• Parameter complexity increases when combined with differential evolution operators • Performance degrades on very large instances (>200 jobs) due to search space expansion

No.	Author / Year	Research Title	Advantages of PSO Usage	Disadvantages of PSO Usage
			dynamic scheduling environments	• Prone to premature convergence on highly multi-modal solution landscapes
2	Li et al. (2021)	PSO+LOA: Hybrid Constrained Optimization for Scheduling Scientific Workflows in the Cloud	• PSO-LOA combination enhances global exploration, avoiding poor local optima • Reduces makespan and execution cost compared to standalone PSO and list-based heuristics • Well-suited for heterogeneous cloud environments with complex resource constraints	• Computational overhead increases due to dual-algorithm integration • Dual parameter sets (PSO + LOA) require more complex and time-consuming tuning • Not evaluated on dynamic cloud scenarios with high workload fluctuations
3	Kaya et al. (2021) [9]	Solution for Flow Shop Scheduling Using Chaotic Hybrid Firefly and PSO with Improved Local Search	• Chaotic maps increase population diversity, effectively escaping local optima • Improved local search accelerates convergence on Taillard standard benchmarks • Outperforms basic PSO and NEH heuristic across most tested instances	• Chaotic hyperparameters are difficult to optimise systematically • Performance is sensitive to initialisation quality on very large problem instances • Higher computational overhead from combining three components (PSO, Firefly, Local Search)
4	Lian et al. (2022) [6]	A Self-Adaptive PSO with Gaussian Perturbation for Stochastic Assembly Line Balancing	• Inertia weight adapts automatically without manual parameter tuning • Gaussian perturbation improves robustness in stochastic scheduling environments • Consistently produces near-optimal solutions across 100 stochastic scenarios	• Gaussian perturbation increases computational cost per iteration • Less effective when stochastic distributions are highly asymmetric (heavy-tailed) • Difficult to directly apply to scheduling problems with strict sequencing constraints
5	Arora & Banyal (2022)	A Particle Grey Wolf Hybrid Algorithm for Workflow Scheduling in Cloud Computing	• PSO-GWO synergy strengthens exploitation capability without sacrificing global exploration • Achieves greater makespan reduction than standalone PSO and GA • Robust against dynamic workload variations in cloud environments	• Number of parameters requiring tuning increases significantly • Higher implementation complexity compared to single-algorithm approaches • Longer execution time due to grey wolf hierarchy mechanism overhead
6	Rosati et al. (2021)	Multi-Objective PSO for Surgical Scheduling: Patient Waiting Time and Resource Utilisation	• Simultaneously optimises multiple competing objectives (waiting time, utilisation, preferences) • Reduces average patient waiting time by 23% compared to conventional hospital scheduling • Easily accommodates new constraints without altering the core PSO structure	• Pareto front quality is highly dependent on external archive size and update strategy • Sensitive to objective function weight selection under certain scheduling conditions • Not tested on hospital scenarios with unexpected disruptions (e.g., operation cancellations)

No.	Author / Year	Research Title	Advantages of PSO Usage	Disadvantages of PSO Usage
7	Pang et al. (2023)	Privacy-Preserving PSO for Secure Cloud Task Scheduling	• Differential privacy mechanism allows PSO to operate on sensitive data without exposure • Maintains effective load balancing while preserving data security • Demonstrates acceptable trade-off between privacy protection and scheduling performance	• Added differential privacy noise can reduce PSO convergence accuracy • Computational overhead increases due to encryption at each particle evaluation • Performance degrades under very high privacy sensitivity requirements (large noise budget)
8	Li et al. (2023)	PSO-Enhanced Deep Reinforcement Learning for Electric Vehicle Charging Scheduling	• PSO-DRL integration produces adaptive schedules responsive to real-time demand variations • Significantly minimises peak grid load while satisfying user time-window constraints • PSO accelerates DRL training convergence by optimising hyperparameters automatically	• Extremely high computational requirements due to deep reinforcement learning training • Requires large historical demand datasets for effective initial model training • PSO performance as a DRL optimiser depends heavily on the quality of the reward function
9	Heidari et al. (2024)	Optimisation of Carbon Emission in Integrated Machine Scheduling and Vehicle Routing Using MOPSO and NSGA-II	• MOPSO efficiently explores a three-dimensional solution space (time, cost, carbon emission) • Generated Pareto front dominates NSGA-II solutions on most tested instances • Integrated approach simultaneously bridges operational and environmental objectives	• Combined problem complexity (scheduling + routing) increases computation time exponentially • MOPSO solutions are sensitive to Pareto archive size and update strategies • Validation limited to specific manufacturing scenarios, not yet generalised to other industries
10	Wang et al. (2025)	Discrete Differential Evolution-PSO (D-DEPSO) for Energy-Saving Flexible Job Shop Scheduling	• Discrete DE-PSO combination effectively handles job permutation representation • Outperforms benchmark algorithms on standard FJSS instances in energy consumption reduction • TPOF (Tripartite Precedence Operation Fusion) mechanism enhances population diversity	• Dual parameters (PSO + DE) require Taguchi calibration process for optimal configuration • Discrete encoding complexity increases implementation overhead compared to continuous PSO • Not extensively tested on FJSS problems with machine breakdowns or dynamic job arrivals

8.1. Trends in the Application of PSO for Scheduling Problems

Table 3 presents a comparison of ten representative studies on the application of Particle Swarm Optimization (PSO) for scheduling problems published between 2021 and 2025. The reviewed studies demonstrate a clear evolution in PSO research, from conventional scheduling optimization toward more complex and intelligent scheduling environments. Recent applications are no longer limited to manufacturing systems but have expanded to cloud computing, healthcare, electric vehicle charging, and integrated scheduling and routing problems. A significant trend observed in the reviewed studies is the increasing adoption of hybrid PSO algorithms rather than the standard PSO. Most researchers combine PSO with other optimization techniques, such as Differential Evolution, Grey Wolf Optimizer, Firefly Algorithm, Local Search, Gaussian Perturbation, and Deep Reinforcement Learning, to overcome the limitations of conventional PSO. These hybrid approaches improve global exploration, prevent premature convergence, and enhance solution quality when solving complex scheduling problems.

8.2. Advantages of PSO in Scheduling Optimization

The comparison of previous studies indicates several common advantages of applying PSO to scheduling problems. First, PSO consistently improves scheduling performance by minimizing important optimization objectives such as makespan, tardiness, energy consumption, execution cost, waiting time, and carbon emissions. This demonstrates that PSO is highly flexible and can accommodate both single-objective and multi-objective optimization problems. Second, many studies reported that PSO provides faster convergence than conventional optimization algorithms. Hybrid approaches such as PSO-Differential Evolution, PSO-Grey Wolf Optimizer, and Chaotic Hybrid Firefly-PSO further enhance convergence speed while maintaining solution quality. These findings indicate that integrating PSO with complementary optimization techniques significantly improves search efficiency. Third, the reviewed studies show that PSO performs well in dynamic and complex environments. Applications in cloud workflow scheduling, healthcare scheduling, electric vehicle charging, and integrated manufacturing systems demonstrate that PSO can effectively handle multiple constraints, heterogeneous resources, and conflicting optimization objectives. Furthermore, recent developments have extended PSO to support intelligent scheduling through integration with deep reinforcement learning and privacy-preserving mechanisms.

8.3. Limitations of Existing PSO Approaches

Despite its promising performance, several limitations consistently appear across the reviewed studies. One of the most frequently reported challenges is premature convergence, where particles become trapped in local optima, particularly in large-scale and highly complex scheduling problems. This issue motivates researchers to introduce hybrid optimization strategies and adaptive parameter adjustment techniques. Another common limitation is the increasing computational complexity associated with hybrid PSO algorithms. Combining PSO with additional optimization techniques generally improves solution quality but simultaneously increases execution time, algorithm complexity, and computational overhead. Parameter tuning also remains an important issue. Most hybrid algorithms require multiple control parameters, making the optimization process more difficult and reducing algorithm robustness across different scheduling environments. Furthermore, several studies reported that their proposed methods were validated only on benchmark datasets or specific industrial scenarios, limiting the generalizability of the results to broader real-world applications.

8.4. Emerging Research Trends

The reviewed literature reveals several emerging research trends in PSO-based scheduling. Recent studies increasingly focus on multi-objective optimization, where scheduling decisions simultaneously consider economic, operational, and environmental objectives. Energy-aware scheduling, carbon emission reduction, resource utilization, and sustainability have become major research topics, reflecting the growing importance of green manufacturing and sustainable production systems. Another emerging trend is the integration of PSO with artificial intelligence techniques, particularly Deep Reinforcement Learning and privacy-preserving optimization methods. These approaches enable adaptive scheduling decisions under dynamic and uncertain environments while maintaining optimization performance.

8.5. Critical Synthesis and Research Implications

Overall, the reviewed studies demonstrate that Particle Swarm Optimization remains one of the most effective metaheuristic algorithms for solving scheduling problems. However, the evolution of PSO research clearly indicates a transition from conventional optimization toward hybrid and intelligent optimization approaches. Although hybrid PSO algorithms generally achieve superior optimization performance, they also introduce greater computational complexity and parameter tuning challenges. The comparison further shows that no single PSO variant consistently outperforms all others across different scheduling domains. Instead, the effectiveness of a particular PSO variant depends on the characteristics of the scheduling problem, optimization objectives, and environmental constraints. Consequently, future research should focus on developing adaptive PSO algorithms capable of automatically adjusting parameters, improving scalability for large-scale scheduling problems, and integrating machine learning techniques to support intelligent decision-making in dynamic production environments.

9. Conclusion

This systematic literature review has synthesized 35 peer-reviewed studies on Particle Swarm Optimization for scheduling problems published between 2004 and 2024. The review demonstrates that PSO has been successfully applied across a diverse range of scheduling domains including job shop and flow shop scheduling, cloud computing task allocation, healthcare and nurse rostering, transportation logistics, and educational timetabling. The PSO family of algorithms has been substantially extended beyond the original standard formulation to include inertia-weight variants, hybrid approaches combining PSO with genetic algorithms and simulated annealing, multi-objective formulations for Pareto-optimal scheduling, and adaptive self-tuning mechanisms for dynamic environments.

Evidence consistently indicates that hybrid PSO variants outperform standard PSO in solution quality, while adaptive PSO variants demonstrate superior robustness in dynamic scheduling contexts. Multi-objective PSO provides competitive Pareto front quality relative to NSGA-II and other established multi-objective metaheuristics [21]. Key open challenges include premature convergence for large-scale instances, parameter sensitivity requiring systematic automated tuning, and limited scalability analysis. The integration of PSO with machine learning and deep learning represents the most promising frontier for future research, potentially enabling adaptive, data-driven scheduling optimization.

This review provides researchers and practitioners with a comprehensive, structured overview of the state-of-the-art in PSO-based scheduling, enabling informed selection of algorithmic variants, identification of research gaps, and guidance for future algorithmic development.

References

[1] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th ed. Springer, 2016.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: W. H. Freeman, 1979.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimisation," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[4] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimisation: An overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[5] D. Y. Sha and C. Y. Hsu, "A hybrid particle swarm optimisation for job shop scheduling problem," *Computers & Industrial Engineering*, vol. 51, no. 4, pp. 791–808, 2006.

[6] Z. Lian, X. Gu, and B. Jiao, "A similar particle swarm optimisation algorithm for permutation flowshop scheduling to minimise makespan," *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 773–785, 2006.

[7] J. Ding, S. Schulz, L. Shen, U. Buscher, and Z. Lü, "Energy-aware scheduling in flexible flow shops with hybrid particle swarm optimization," *Computers & Operations Research*, vol. 125, Art. no. 105088, Jan. 2021.

[8] Y. Wang and X. Zuo, "An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 1079–1094, 2021.

[9] S. Kaya, A. Gümüşçü, İ. B. Aydilek, T. Saraç, and A. Allahverdi, "Solution for flow shop scheduling problems using chaotic hybrid firefly and particle swarm optimization algorithm with improved local search," *Soft Computing*, vol. 25, no. 9, pp. 7143–7154, 2021.

[10] S. Nabi, M. Ahmad, M. Ibrahim, and H. Hamam, "AdPSO: Adaptive PSO-based task scheduling approach for cloud computing," *Sensors*, vol. 22, no. 3, Art. no. 920, 2022.

[11] A. I. Awad, M. M. Fouda, M. M. Khashaba, E. R. Mohamed, and K. M. Hosny, "Efficient workflow scheduling in fog-cloud collaboration using a hybrid IPSO-GWO algorithm," *Scientific Reports*, vol. 15, Art. no. 13591, 2025.

[12] Y. Shi and R. Eberhart, "A modified particle swarm optimiser," in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 69–73, 1998.

[13] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[14] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, *Handbook on Scheduling: From Theory to Practice*. Cham, Switzerland: Springer, 2019.

[15] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLOS Medicine*, vol. 6, no. 7, Art. no. e1000097, 2009.

[16] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimisation algorithm for makespan and total flowtime minimisation in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930–1947, 2007.
[17] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimisation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 6, pp. 1362–1381, 2009.
[18] C. Akkan and A. Gulcu, "A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem," *Computers & Operations Research*, vol. 90, pp. 22–32, Feb. 2018.
[19] Y. Xie et al., "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud-edge environment," *Future Generation Computer Systems*, vol. 97, pp. 361–378, 2019.
[20] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, Jun. 2004.
[21] Q. K. Pan, M. F. Tasgetiren, and Y. C. Liang, "A discrete differential evolution algorithm for the permutation flowshop scheduling problem," *Computers & Industrial Engineering*, vol. 55, no. 4, pp. 795–816, 2008.
[22] M. Farid, R. Latip, M. Hussin, and N. A. W. Abdul Hamid, "A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing," *Symmetry*, vol. 12, no. 4, Art. no. 551, Apr. 2020.
[23] J. Xiong, L. Xing, and Y. Chen, "Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns," *International Journal of Production Economics*, vol. 141, no. 1, pp. 112–126, 2012.
[24] J. Tang, G. Zhang, B. Lin, and B. Zhang, "A Hybrid Algorithm for Flexible Job-Shop Scheduling Problem," *Procedia Engineering*, vol. 15, pp. 3678–3683, 2011.
[25] N. Singh and S. B. Singh, "Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance," *Journal of Applied Mathematics*, vol. 2017, Art. ID 2030489, 2017.
[26] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimisation scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, vol. 3, pp. 2687–2699, 2015.
[27] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 400–407, 2010.
[28] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126–140, 2013.
[29] M. Dehghani, E. Trojovská, and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimisation problems on the base of simulation of driving training process," *Scientific Reports*, vol. 12, Art. no. 9924, 2022.
[30] Y. J. Gong, J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung, and Y. Shi, "Optimising the vehicle routing problem with time windows: A discrete particle swarm optimisation approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 2, pp. 254–267, Mar. 2012.
[31] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz, "Planning, operation, and control of bus transit systems: A literature review," *Transportation Research Part B: Methodological*, vol. 77, pp. 38–75, 2015.
[32] G. N. Beligiannis, C. N. Moschopoulos, G. P. Kaperonis, and S. D. Likothanassis, "Applying evolutionary computation to the school timetabling problem: The Greek case," *Computers & Operations Research*, vol. 35, no. 4, pp. 1265–1280, Apr. 2008.
[33] C. Nothegger, A. Mayer, A. Chwatal, and G. R. Raidl, "Solving the post enrolment course timetabling problem by ant colony optimisation," *Annals of Operations Research*, vol. 194, no. 1, pp. 325–339, 2012.
[34] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.
[35] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, Feb. 2002.