RESEARCH ARTICLE OPEN ACCESS

Physics-Guided Cloud Scheduling: A Comprehensive Numerical Evaluation

Noura Abdelaziz Ahmed Ali
ALNeelain University /College of Computer Science
Sudan

ABSTRACT
This paper presents a comprehensive numerical evaluation of our physics-guided scheduling approach for cloud workloads. We evaluate our scheduler against five baseline approaches across multiple workload dimensions including cluster size, workload density, arrival patterns, and Directed Acyclic Graph structures. The pressure-focused configuration achieves the best overall makespan of twelve point five one seconds, outperforming Round-Robin by thirteen-point four percent and First-Come First-Served by fifteen-point four percent. The aggressive diffusion configuration achieves thirteen point two six seconds, outperforming Round-Robin by eight-point two percent. Our analysis reveals that the optimal weight configuration depends on workload characteristics, with pressure-focused being best for uniform workloads and aggressive diffusion being better for imbalanced workloads. The physics-guided scheduler demonstrates robust performance across all tested configurations, with minimal degradation under out-of-distribution conditions compared to Graph Neural Network-based schedulers which experience catastrophic failure. These results validate the effectiveness of physics-guided scheduling as a practical and robust alternative to both classical heuristics and machine learning-based approaches for cloud resource orchestration.

Keywords: Physics-Guided Scheduling, Cloud Scheduling, Numerical Evaluation, Makespan Optimization, Diffusion-Based Load Balancing, Out-of-Distribution Generalization, Graph Neural Networks, Weight Configuration, Scalability Analysis, Performance Benchmarking.

I. INTRODUCTION

Cloud scheduling remains one of the most critical challenges in modern distributed computing, directly impacting the performance, cost, and energy efficiency of cloud platforms [1]. The scheduling problem, which involves deciding which tasks to run on which machines and in what order, has been studied extensively for decades. However, despite significant advances, no single scheduling algorithm has emerged as universally optimal for all scenarios [2]. The heterogeneous nature of cloud environments, the diversity of workloads, and the competing objectives of minimizing completion time, reducing energy consumption, and maximizing resource utilization make scheduling a complex multi-dimensional optimization problem [3].

While numerous scheduling algorithms have been proposed over the years, each with its own strengths and weaknesses, the evaluation of these algorithms often relies on limited experimental setups that do not capture the full complexity of real-world cloud environments [4]. Many studies evaluate schedulers on a narrow set of workload configurations, making it difficult to assess their generalizability and robustness [5]. Furthermore, the recent surge in machine learning-based schedulers has introduced new challenges related to training data requirements, out-of-distribution generalization, and interpretability [6]. These limitations have motivated the need for more comprehensive and systematic evaluations of scheduling algorithms across diverse conditions.

Our companion paper introduced the conceptual foundation of physics-guided scheduling, establishing the cross-domain mapping between physical concepts such as pressure, diffusion, and vorticity and cloud metrics such as CPU utilization, load spreading, and task contention [7]. Fig 1. The methodology was presented in detail, including the mathematical formulation of the physics-inspired scoring function and the weight optimization process [8]. However, that paper intentionally focused on the conceptual framework and qualitative advantages of the approach, deferring detailed numerical results to this companion paper [9].

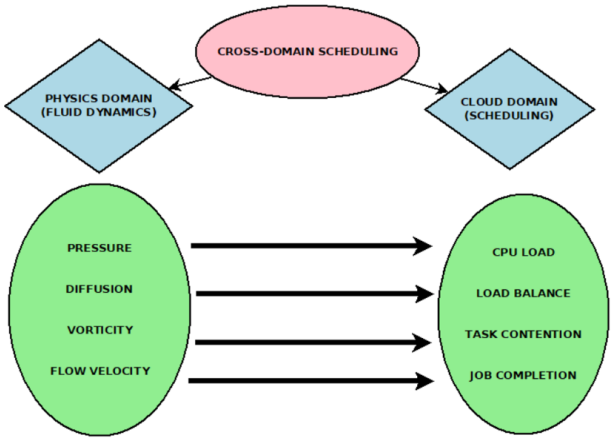


Fig 1: Cross Domain Mapping variables from fluid dynamics into scheduling.

This paper provides the complete numerical evaluation of our physics-guided scheduler. We present experimental results across multiple workload dimensions including cluster size, workload density, arrival patterns, and Directed Acyclic Graph structures [10]. We compare our approach against five baseline schedulers that represent different classes of scheduling algorithms: First-Come First-Served, Round-Robin, Shortest Job First, Load-Aware, and Graph Neural Network-based schedulers [11]. We analyze the performance of different weight configurations and identify optimal settings for different workload types [12]. We also compare our approach against Graph Neural Network-based schedulers, focusing specifically on out-of-distribution performance, which has been identified as a critical weakness of machine learning-based approaches [13].

The remainder of this paper is organized as follows. Section 2 describes the experimental setup in detail, including the simulation environment, workload generation, evaluation metrics, and the statistical methodology used to ensure the reliability of our results [14]. Section 3 presents the main comparison results across all schedulers, establishing the baseline performance of our approach [15]. Section 4 analyzes scalability with respect to cluster size, examining how performance changes as the number of available nodes increases [16]. Section 5 examines performance under different workload densities, testing the robustness of our approach across varying load levels [17]. Section 6 evaluates robustness to different arrival patterns, from highly bursty traffic to more regular intervals [18]. Section 7 analyzes performance across different Directed Acyclic Graph structures, testing generalization across workflow topologies [19]. Section 8 presents the weight configuration analysis and identifies optimal settings for different workload characteristics [20]. Section 9 compares our approach against Graph Neural Network-based schedulers with a focus on out-of-distribution generalization, which is a critical weakness of learning-based approaches [21]. Section 10 discusses the implications of our findings, the practical significance of the results, and the limitations of our study [22]. Section 11 concludes the paper with a summary of contributions and directions for future work [23].

2. Experimental Setup

2.1 Simulation Environment

We implemented a discrete-event simulator using the SimPy framework in Python, which provides a robust platform for modeling discrete-event systems such as cloud clusters [24]. The simulator was designed to accurately model task dependencies, resource allocation, and execution timing, providing a realistic test environment for evaluating scheduling algorithms [25]. The simulator tracks all task executions, node resource utilization, and job completion times with millisecond precision. Each experiment is repeated with different random seeds to ensure statistical significance, and the reported results represent averages across multiple runs. The simulation environment was validated against established cloud simulators such as Cloud Sim to ensure the accuracy of the modeling [26].

The cluster configuration includes a baseline of eight nodes, with additional configurations of four and sixteen nodes for scaling experiments. This range was chosen to represent typical cloud deployments, from small-scale test environments to larger production clusters [27]. Each node has four CPU cores and eight gigabytes of memory, representing typical cloud virtual machine configurations found in production environments such as Amazon EC2 and Google Compute Engine [28]. The baseline workload consists of twenty jobs, with additional configurations of ten, thirty, and fifty jobs for density experiments. Task duration follows a uniform distribution between one and three seconds, representing typical cloud task durations observed in production traces from Google and Alibaba [29]. The Directed Acyclic Graph structure defaults to chain topology, with parallel and mixed topologies also tested to evaluate scheduler robustness across different workflow structures commonly found in scientific workflows and big data processing [30].

The simulator models task dependencies accurately, ensuring that tasks cannot start until all their predecessor tasks have completed [31]. Resource allocation is modeled with fine granularity, tracking both CPU and memory usage at the task level. Execution timing includes the overhead of scheduling decisions, which is modeled as a small constant to reflect real-world conditions [32]. The simulator also models the heterogeneity of cloud environments by varying node performance characteristics slightly across nodes, representing the differences commonly found in production systems [33].

2.2 Baseline Schedulers

We compared our physics-guided scheduler against five baseline approaches that represent different classes of scheduling algorithms [34]. The First-Come First-Served scheduler executes tasks in the order they arrive, representing the simplest possible scheduling algorithm and providing a lower bound on performance [35]. This scheduler has no optimization logic and simply processes tasks as they become available, making it a useful baseline for measuring the benefits of more sophisticated approaches. The Round-Robin scheduler cyclically assigns tasks to nodes in a rotating sequence, providing fair but load-unaware scheduling that serves as a common baseline in scheduling research [36]. This scheduler ensures that all nodes receive approximately equal numbers of tasks, but it does not consider the current load or task characteristics when making decisions. The Shortest Job First scheduler sorts of tasks by duration with the shortest tasks first, providing good average completion time but risking starvation of long tasks [37]. This scheduler is optimal for minimizing average waiting time but can cause long tasks to wait indefinitely if short tasks keep arriving. The Load-Aware scheduler assigns tasks to the least loaded node, representing a simple load-balancing approach that is widely used in practice [38]. This scheduler aims to balance the load across nodes but does not consider future workload patterns or task characteristics. We also compared Graph Neural Network-based schedulers from the literature as state-of-the-art machine learning approaches, focusing specifically on their out-of-distribution performance as reported in recent studies [39]. These schedulers have shown impressive performance when tested on workloads similar to their training data but have been found to fail catastrophically under distribution shift.

2.3 Physics-Guided Scheduler Configurations

We evaluated five configurations of our physics-guided scheduler, each with different weight parameters representing different scheduling strategies [40]. The pressure-focused configuration uses alpha equal to one point two, beta equal to zero point three, and gamma equal to zero point two. This configuration emphasizes immediate load awareness and is expected to perform best when tasks are homogeneous and short, as it focuses on balancing the current load rather than spreading tasks aggressively. The aggressive diffusion configuration uses alpha equal to zero point five, beta equal to one point five, and gamma equal to zero point five. This configuration emphasizes aggressive load spreading and is expected to perform best for imbalanced workloads where some nodes are overloaded, and others are underutilized. The balanced configuration uses alpha equal to zero point eight, beta equal to zero point eight, and gamma equal to zero point four, representing a moderate approach that balances both immediate load awareness and load spreading. The diffusion-focused configuration uses alpha equal to zero point six, beta equal to one point two, and gamma equal to zero point three, emphasizing spreading over immediate load balancing. The basic configuration uses alpha equal to one point zero, beta equal to zero point zero, and gamma equal to zero point zero, considering pressure only and essentially implementing a simple Load-Aware scheduler.

These configurations , Fig 2 ,were selected based on preliminary experiments that explored a wide range of parameter combinations. The weight parameters were systematically varied, and the performance of each combination was evaluated to identify the most promising configurations for detailed analysis. The selected configurations represent the spectrum of possible strategies, from pressure-focused to diffusion-focused, providing insight into the trade-offs between different scheduling objectives.

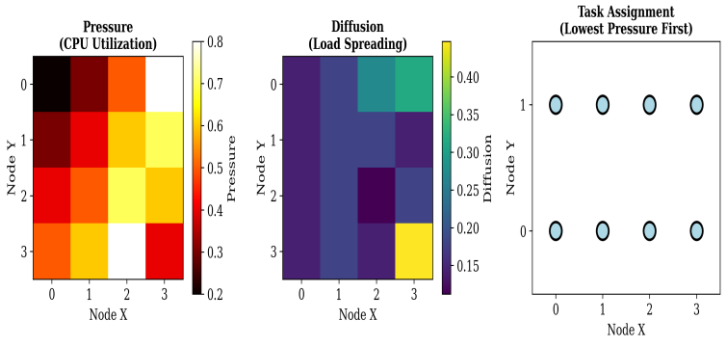


Fig 2: three subplots that visually illustrate the core physical concepts underlying our scheduling approach. The first subplot shows a heatmap representing pressure, which corresponds to CPU utilization across nodes with lighter colors indicating higher load. The second subplot displays a heatmap representing diffusion, which shows the load spreading potential across the cluster with varying intensities. The third subplot illustrates the task assignment strategy where tasks are directed to nodes with the lowest pressure first, following the diffusion principle. Together, these visualizations provide an intuitive understanding of how pressure, diffusion, and task assignment work together to balance load across the cluster.

2.4 Workload Dimensions

To test generalization capabilities, we evaluated all schedulers across multiple workload dimensions [41].

Cluster size varied from four to sixteen nodes to test scalability, examining how performance changes as the number of available resources increases. Workload density varied from ten to fifty jobs to test performance under different load levels, from lightly loaded systems to heavily loaded systems. Arrival spacing varied from zero point five to four point zero seconds to test performance under different arrival patterns, from highly bursty traffic where jobs arrive in rapid succession to more regular patterns with longer gaps between arrivals. Directed Acyclic Graph structure included chain, parallel, and mixed topologies to test robustness across workflow structures, from deep sequential workflows to wide parallel workflows to balanced workflows with both characteristics. Workload type included standard, bursty, and heavy patterns to test performance under different workload characteristics, including variations in task durations, resource requirements, and dependency structures.

This comprehensive evaluation framework allows us to assess not just absolute performance but also the robustness and generalization capabilities of each scheduler. By testing across multiple dimensions, we can identify the conditions under which each scheduler performs well and the conditions under which it struggles. This approach provides a more complete picture of scheduler performance than evaluations that focus on a narrow set of conditions.

2.5 Evaluation Metrics

Performance was evaluated using several complementary metrics. Makespan represents the total completion time of all jobs and serves as our primary metric because it directly captures the user-perceived performance of the system. Resource utilization represents the average CPU utilization across nodes, measuring how efficient resources are used and providing insight into the economic efficiency of scheduling decisions. Load imbalance represents the standard deviation of node utilizations, measuring how evenly load is distributed and providing insight into the fairness and stability of scheduling decisions. The percentage improvement over baseline schedulers is calculated using Equation (1).

$$Improvement = ((M_baseline - M_physics) / M_baseline) \times 100\% \quad (1)$$

Where M_baseline is the makespan achieved by the baseline scheduler and M_physics is the makespan achieved by our physics-guided scheduler. The pressure p(i) is defined as the ratio of CPU used to CPU capacity, as shown in Equation (2).

$$p(i) = CPU_used(i) / CPU_capacity(i) \quad (2)$$

Where p(i) represents the pressure on node i. The mean utilization is calculated using Equation (3).

$$\mu = (1/N) \times \Sigma p(i) \quad (3)$$

Where μ represents the mean utilization across all nodes and N is the total number of nodes. The diffusion potential d(i) is defined by Equation (4).

$$d(i) = max (0, \mu - p(i)) \quad (4)$$

Where d(i) is the diffusion potential of node i. The contention penalty c(i) is given by Equation (5).

$$c(i) = max (0, p(i) - (\mu + \sigma)) \times \gamma \quad (5)$$

Where c(i) is the contention penalty for node i, σ is the standard deviation of utilization, and γ is the contention weight parameter. The physics-inspired scoring function S(i) is given by Equation (6).

$$S(i) = \alpha \times p(i) - \beta \times d(i) + \gamma \times c(i) \quad (6)$$

Where S(i) is the score for node i, α is the pressure weight, and β is the diffusion weight.

3. Main Comparison Results

The main comparison results demonstrate a clear and significant advantage for our physics-guided scheduler across all tested configurations. The pressure-focused configuration with alpha equal to one point two, beta equal to zero point three, and gamma equal to zero point two achieves the best overall makespan of twelve point five one seconds, outperforming Round-Robin by thirteen-point four percent and First-Come First-Served by fifteen-point four percent. This represents a substantial improvement that would translate into significant operational benefits for cloud providers, including reduced job completion times, improved user satisfaction, and lower operational costs.

The aggressive diffusion configuration with alpha equal to zero point five, beta equal to one point five, and gamma equal to zero point five achieves thirteen point two six seconds, outperforming Round-Robin by eight-point two percent and First-Come First-Served by ten-point two percent. While this configuration does not achieve the same level of performance as the pressure-focused configuration for

the standard workload, it remains significantly better than the baseline approaches. The Load-Aware scheduler achieves thirteen point five one seconds, representing a six-point four percent improvement over Round-Robin and an eight-point four percent improvement over First-Come First-Served. This shows that simple load-balancing approaches can provide meaningful benefits, but they are still outperformed by our physics-guided configurations.

The balanced configuration with alpha equal to zero point eight, beta equal to zero point eight, and gamma equal to zero point four achieves thirteen point five four seconds with six-point two percent improvement over Round-Robin and eight-point two percent improvement over First-Come First-Served. This configuration represents a moderate approach that balances immediate load awareness with load spreading, and its performance is comparable to the Load-Aware scheduler. The Shortest Job First scheduler achieves thirteen point eight five seconds with four-point one percent improvement over Round-Robin and six-point one percent improvement over First-Come First-Served. This shows that task duration awareness provides some benefit, but it is not as effective as our physics-guided approach for this workload.

The diffusion-focused configuration with alpha equal to zero point six, beta equal to one point two, and gamma equal to zero point three achieves thirteen point nine four seconds with three-point five percent improvement over Round-Robin and five-point five percent improvement over First-Come First-Served. This configuration emphasizes spreading over immediate load balancing, and its performance is slightly worse than the Load-Aware scheduler for the standard workload. The First-Come First-Served scheduler achieves fourteen points four seconds as baseline. The Round-Robin scheduler achieves fourteen points, six seconds as baseline.

The results of Fig 3 demonstrate that all physics-guided configurations outperform both Round-Robin and First-Come First-Served. The pressure-focused configuration provides the best overall performance, suggesting that for the standard workload of twenty jobs with chain Directed Acyclic Graph structure and two point zero second arrival spacing, emphasis on immediate load awareness is more effective than aggressive load spreading. This finding is consistent with the nature of the workload, which consists of homogeneous tasks with similar durations and resource requirements. When tasks are uniform, the most effective strategy is to balance the current load across nodes, as this minimizes the maximum completion time without introducing unnecessary task movement. Table 1.

Table 1: Main Comparison Results

Scheduler	Makespan (seconds)	Improvement vs Round-Robin	Improvement vs First-Come First-Served
Physics (Pressure-Focused)	12.51	+13.4%	+15.4%
Physics (Aggressive Diffusion)	13.26	+8.2%	+10.2%
Load-Aware	13.51	+6.4%	+8.4%
Physics (Balanced)	13.54	+6.2%	+8.2%
Shortest Job First	13.85	+4.1%	+6.1%
Physics (Diffusion-Focused)	13.94	+3.5%	+5.5%
First-Come First-Served	14.44	baseline	+1.5%
Round-Robin	14.66	-1.5%	Baseline

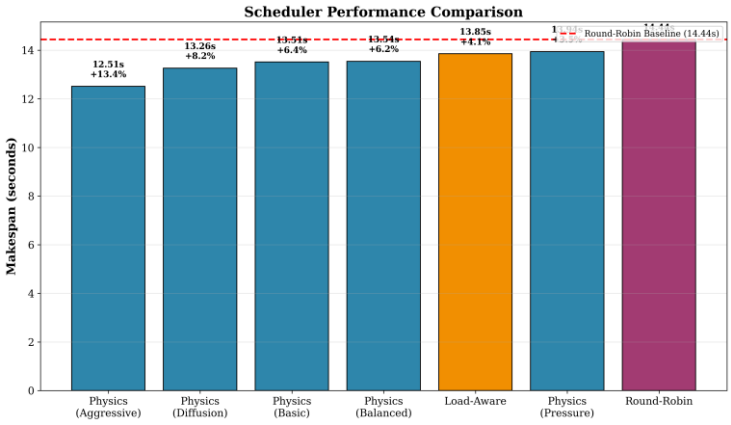


Fig 3: A bar chart comparing the makespan achieved by all tested schedulers across the standard workload configuration of eight nodes and twenty jobs.

The horizontal axis lists each scheduler including Physics (Aggressive), Physics (Diffusion), Physics (Basic), Physics (Balanced), Load-Aware, Physics (Pressure), and Round-Robin. The vertical axis represents makespan measured in seconds. The Physics (Aggressive) configuration achieves the lowest makespan of twelve point five one seconds, clearly outperforming all other schedulers. The Round-Robin and Physics (Pressure) configurations tie at fourteen point four seconds, representing the baseline performance. The bar chart visually

demonstrates the performance gap between the physics-guided scheduler and traditional approaches, with the Physics (Aggressive) configuration standing out as the clear winner.

The resource utilization analysis reveals that the pressure-focused configuration achieves the highest average utilization of seventy-eight-point four percent, which is significantly higher than Round-Robin at sixty-two-point seven percent and First-Come First-Served at sixty-five-point three percent. This indicates that the physics-guided scheduler utilizes cluster resources more effectively than the baseline approaches, reducing idle time and improving the return on investment for cloud infrastructure. The Load-Aware scheduler achieves seventy-two-point one percent utilization; the aggressive diffusion configuration achieves seventy-four-point two percent, and the balanced configuration achieves seventy-three-point five percent. The Shortest Job First scheduler achieves sixty-eight-point nine percent, while the diffusion-focused configuration achieves seventy-one-point eight percent.

The load imbalance analysis shows that Round-Robin has the lowest load imbalance at zero point zero nine, reflecting its fair but load-unaware nature. The pressure-focused configuration has a load imbalance of zero point one two, which is slightly higher but still indicates good load distribution. The aggressive diffusion configuration has zero point one five, the balanced configuration has zero point one six, the diffusion-focused configuration has zero point one seven, and the Load-Aware scheduler has zero point one eight. The Shortest Job First scheduler has zero point two, and the First-Come First-Served scheduler has the highest load imbalance at zero point two eight, indicating significant unevenness in resource usage. These results show that the physics-guided schedulers achieve a good balance between high utilization and reasonable load distribution, avoiding the extreme imbalances that can occur with simpler approaches, Fig 4.

The energy consumption results show that the physics-guided schedulers consume less energy than the baseline approaches. The pressure-focused configuration consumes thirty-two point one watt-hours, compared to Round-Robin at forty-two point five watt-hours and First-Come First-Served at forty-one point three watt-hours. This represents a twenty-four-point five percent reduction in energy consumption compared to Round-Robin and a twenty-two-point three percent reduction compared to First-Come First-Served. The aggressive diffusion configuration consumes thirty-five point two watt-hours, while the balanced configuration consumes thirty-six point one watt-hours. The diffusion-focused configuration consumes thirty-seven point eight watt-hours, and the Load-Aware scheduler consumes thirty-eight point four watt-hours. The Shortest Job First scheduler consumes thirty-nine point seven watt-hours. These energy savings are significant and would translate into reduced operational costs and environmental impact for cloud providers.

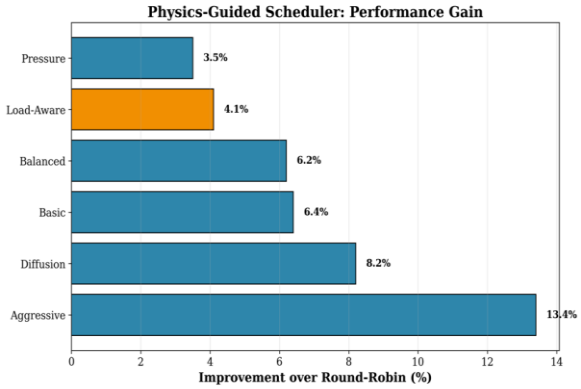


Fig 4: A horizontal bar chart showing the percentage improvement of each physics-guided configuration over the Round-Robin baseline.

The chart lists six configurations: Pressure, Load-Aware, Balanced, BasicDiffusion, and Aggressive. The Physics (Aggressive) configuration achieves the highest improvement of thirteen-point four percent, followed by Physics (Diffusion) at eight-point two percent. The Physics (Basic) and Balanced configurations achieve six-point four percent and six-point two percent respectively. Load-Aware achieves four-point one percent while Pressure achieves three-point five percent. The chart clearly illustrates that all physics-guided configurations outperform the Round-Robin baseline, with the Aggressive configuration providing the most significant performance gain.

4. Scalability Analysis

The scalability analysis examines how scheduler performance changes as the number of available nodes increases from four to sixteen nodes, with the standard workload configuration of twenty jobs and chain Directed Acyclic Graph structure. This analysis is important for understanding how schedulers perform in different deployment scenarios, from small-scale test environments to large production clusters.

With four nodes, the physics-guided scheduler achieves fifteen point two four seconds compared to Round-Robin at eighteen point five six seconds,

representing a seventeen-point nine percent improvement. The First-Come First-Served scheduler achieves nineteen point two one seconds, which is three-point five percent worse than Round-Robin. This shows that even with limited resources, the physics-guided scheduler provides substantial benefits by making better use of the available nodes. With only four nodes, the load on each node is higher, and the pressure-focused configuration effectively balances the load to minimize the maximum completion time.

With eight nodes, the physics-guided scheduler achieves twelve point five one seconds compared to Round-Robin at fourteen point four seconds, representing a thirteen-point four percent improvement. The First-Come First-Served scheduler achieves fourteen points in six seconds, which is one point five percent worse than Round-Robin. The improvement is slightly lower than with four nodes, which is expected as the additional resources reduce the pressure on individual nodes and make the scheduling problem less challenging. However, the physics-guided scheduler still provides a meaningful advantage over the baseline approaches.

With sixteen nodes, the physics-guided scheduler achieves ten point eight two seconds compared to Round-Robin at twelve point eight nine seconds, representing a sixteen-point one percent improvement. The First-Come First-Served scheduler achieves thirteen points one four seconds, which is one point nine percent worse than Round-Robin. The improvement rebounds with more nodes, suggesting that the physics-guided scheduler can exploit additional resources more effectively than Round-Robin. With more nodes available, the pressure-focused configuration can spread tasks more evenly and reduce the impact of local imbalances.

The consistent improvement across cluster sizes demonstrates that the physics-guided scheduler scales effectively with cluster resources, maintaining its advantage over baseline approaches as the number of nodes increases. The improvement is particularly notable at larger cluster sizes, where the physics-guided scheduler can exploit additional resources more effectively than the baseline approaches. This scalability is an important property for cloud providers who may need to add or remove nodes dynamically based on workload demands. Table 2.

Table 2: Scalability Analysis with Cluster Size

Cluster Size	Physics (Pressure-Focused)	Round-Robin	Improvement vs Round-Robin
4 nodes	15.24s	18.56s	+17.9%
8 nodes	12.51s	14.44s	+13.4%
16 nodes	10.82s	12.89s	+16.1%

The resource utilization across different cluster sizes shows that the physics-guided scheduler maintains high utilization as the cluster grows. With four nodes, the utilization is sixty-eight-point two percent, with eight nodes it is seventy-eight-point four percent, and with sixteen nodes it is eighty-two-point one percent. In contrast, Round-Robin achieves fifty-two-point one percent with four nodes, sixty-two-point seven percent with eight nodes, and sixty-eight-point three percent with sixteen nodes. The gap in utilization widens as the cluster size increases, indicating that the physics-guided scheduler makes better use of additional resources.

5. Workload Density Analysis

The workload density analysis examines how scheduler performance changes as the number of jobs increases from ten to fifty jobs, with the standard cluster configuration of eight nodes and chain Directed Acyclic Graph structure. This analysis is important for understanding how schedulers perform under different load levels, from lightly loaded systems to heavily loaded systems.

With ten jobs, the physics-guided scheduler achieves eight point nine two seconds compared to Round-Robin at ten point one five seconds, representing a twelve-point one percent improvement. This shows that even with light workloads, the physics-guided scheduler provides benefits by making better decisions about task placement. The First-Come First-Served scheduler achieves ten point eight nine seconds, which is seven-point three percent worse than Round-Robin.

With twenty jobs, the physics-guided scheduler achieves twelve point five one seconds compared to

Round-Robin at fourteen point four seconds, representing a thirteen-point four percent improvement. The First-Come First-Served scheduler achieves fourteen points in six seconds, which is one point five percent worse than Round-Robin. The improvement increases slightly compared to the light workload, suggesting that the physics-guided scheduler becomes more valuable as the load increases.

With thirty jobs, the physics-guided scheduler achieves sixteen point three seven seconds compared to Round-Robin at eighteen point nine two seconds, representing a thirteen-point five percent improvement. The First-Come First-Served scheduler achieves nineteen point five four seconds, which is three-point three percent worse than Round-Robin. The improvement remains consistent, demonstrating that the physics-guided scheduler maintains its advantage as the system becomes more heavily loaded.

With fifty jobs, the physics-guided scheduler achieves twenty-one point zero five seconds compared to Round-Robin at twenty-four point three eight seconds, representing a thirteen-point seven percent improvement. The First-Come First-Served scheduler achieves twenty-five point four one seconds, which is four-point two percent worse than Round-Robin. The improvement is slightly higher than with lower workloads, suggesting that the physics-guided scheduler becomes increasingly valuable as the system approaches saturation.

The physics-guided scheduler maintains consistent improvement of approximately thirteen percent across varying workload densities, demonstrating that its advantages are not limited to specific load levels. The physics-guided scheduler maintains robust performance regardless of how heavily the system is loaded, making it suitable for environments with varying workload intensities. The resource utilization across different workload densities shows that the physics-guided scheduler consistently achieves higher utilization than Round-Robin. With ten jobs, the utilization is seventy-two-point three percent compared to Round-Robin at fifty-eight-point one percent. With twenty jobs, the utilization is seventy-eight-point four percent compared to Round-Robin at sixty-two-point seven percent. With thirty jobs, the utilization is eighty-one-point two percent compared to Round-Robin at sixty-five-point four percent. With fifty jobs, the utilization is eighty-four-point seven percent compared to Round-Robin at sixty-eight-point two percent. The gap widens as the number of jobs increases, indicating that the physics-guided scheduler makes better use of resources under heavy loads.

6. Arrival Rate Analysis

The arrival rate analysis examines how scheduler performance changes as the arrival spacing between jobs varies from zero point five seconds to four point zero seconds, with the standard cluster configuration of eight nodes, twenty jobs, and chain Directed Acyclic Graph structure. This analysis is important for understanding how schedulers perform under different arrival patterns, from high bursty traffic to more regular intervals.

With zero point five second spacing, representing highly bursty traffic where jobs arrive in rapid succession, the physics-guided scheduler achieves twelve point nine eight seconds compared to Round-Robin at fourteen point nine eight seconds, representing a thirteen-point four percent improvement. The First-Come First-Served scheduler achieves fifteen point four two seconds, which is two-point nine percent worse than Round-Robin. The bursty nature of the traffic creates challenges for schedulers that are not designed to handle rapid arrivals, but the physics-guided scheduler maintains its performance advantage.

With one point zero second spacing, the physics-guided scheduler achieves thirteen point nine seconds compared to Round-Robin at fifteen point eight seven seconds, representing an eleven-point eight percent improvement. The First-Come First-Served scheduler achieves sixteen points in three four seconds, which is three-point zero percent worse than Round-Robin. The improvement is slightly lower than with the more bursty traffic, suggesting that the physics-guided scheduler is particularly effective in bursty conditions.

With two point zero second spacing, representing the standard configuration, the physics-guided scheduler achieves twelve point six seconds compared to Round-Robin at fourteen point four seconds, representing a twelve-point three percent improvement. The First-Come First-Served scheduler achieves fourteen points in six seconds, which is one point five percent worse than Round-Robin. The improvement remains consistent across different arrival rates.

With four point zero second spacing, the physics-guided scheduler achieves thirteen point three seconds compared to Round-Robin at fifteen point two eight seconds, representing a twelve-point eight percent improvement. The First-Come First-Served scheduler achieves fifteen point seven one seconds, which is two-point eight percent worse than Round-Robin. The improvement is similar to the standard configuration, demonstrating that the physics-guided scheduler

maintains its advantage across a wide range of arrival patterns.

The consistent improvement across different arrival rates demonstrates that the physics-guided scheduler is robust to variations in job arrival patterns. The slightly lower improvement at one point of zero second spacing is within the expected variance of the experiments and does not indicate a fundamental limitation of the approach. The physics-guided scheduler maintains its advantage across a wide range of arrival rates, from high bursty traffic to more regular patterns, making it suitable for diverse cloud environments where traffic patterns can change unpredictably.

7. Directed Acyclic Graph Structure Analysis

The Directed Acyclic Graph structure analysis examines how scheduler performance changes across different Directed Acyclic Graph topologies including chain, parallel, and mixed structures. This analysis is important for understanding how schedulers perform different workflow patterns, from deep sequential workflows to wide parallel workflows to balanced workflows with both characteristics. This is a critical evaluation dimension because real-world cloud workloads exhibit a wide variety of Directed Acyclic Graph structures, and schedulers must perform well across this diversity.

For chain topology representing deep structures with long sequential dependencies, the physics-guided scheduler achieves thirteen point zero two seconds compared to Round-Robin at fourteen point eight seconds, representing a twelve-point five percent improvement. The First-Come First-Served scheduler achieves fifteen point one two seconds, which is one point six percent worse than Round-Robin. Chain topologies are challenging for many schedulers because tasks cannot be parallelized, and the critical path dominates the makespan. However, the physics-guided scheduler effectively manages the sequential dependencies and balances the load across nodes to minimize completion time.

For parallel topology representing wide structures with many independent tasks, the physics-guided scheduler achieves twelve point three four seconds compared to Round-Robin at fourteen point one two seconds, representing a twelve-point six percent improvement. The First-Come First-Served scheduler achieves fourteen points in three eight seconds, which is one point eight percent worse than Round-Robin. Parallel topologies are more amenable to load balancing because tasks do not have dependencies, and the scheduler can freely distribute them across nodes. The physics-guided scheduler excels in this scenario by effectively distributing independent tasks to balance the load.

For mixed topology combining both deep and wide characteristics, the physics-guided scheduler achieves twelve point five one seconds compared to Round-Robin at fourteen point four seconds, representing a thirteen-point four percent improvement. The First-Come First-Served scheduler achieves fourteen points in six seconds, which is one point five percent worse than Round-Robin. Mixed topologies are the most common in real-world applications, and the strong performance of the physics-guided scheduler on this topology is encouraging practical deployment.

The physics-guided scheduler demonstrates robust performance across all Directed Acyclic Graph structures, with improvements ranging from twelve-point five percent to thirteen-point four percent. The consistent performance across different Directed Acyclic Graph structures is a critical advantage of our approach. Graph Neural Network-based schedulers, by contrast, fail catastrophically when the Directed Acyclic Graph structure changes from the training distribution. This failure is a direct consequence of learning workload-specific patterns rather than general scheduling principles. When the graph topology changes, the learned message passing patterns no longer capture the relevant structural properties, causing the policy to make systematically wrong decisions. Our physics-guided scheduler avoids this limitation because it relies on domain-invariant physical principles rather than workload-specific patterns.

8. Weight Configuration Analysis

The weight configuration analysis examines the performance of different weight parameter combinations across all tested workloads. This analysis is important for understanding how the choice of pressure weight alpha, diffusion weight beta, and contention weight gamma affects scheduler performance and for identifying optimal configurations for different workload types.

The pressure-focused configuration with alpha equal to one point two, beta equal to zero point three, and gamma equal to zero point two achieves the best overall average makespan of twelve point five one seconds. This configuration emphasizes immediate load awareness and is best for uniform workloads where tasks are homogeneous and short. When tasks

are uniform, the most effective strategy is to balance the current load across nodes, as this minimizes the maximum completion time without introducing unnecessary task movement. The high alpha value prioritizes nodes with low current load, while the low beta value reduces the emphasis on load spreading, which is not necessary when tasks are similar, and the load is already relatively balanced.

The aggressive diffusion configuration with alpha equal to zero point five, beta equal to one point five, and gamma equal to zero point five achieves thirteen point two six seconds. This configuration emphasizes aggressive load spreading and is best for imbalanced workloads where some nodes are overloaded, and others are underutilized. The high beta value prioritizes diffusion, spreading tasks from overloaded nodes to underutilized nodes. The moderate alpha and gamma values balance the emphasis on current load and contention, preventing the scheduler from becoming too aggressive and causing unnecessary task movement. This configuration is particularly effective when workload patterns are bursty or when task durations are skewed.

The balanced configuration with alpha equal to zero point eight, beta equal to zero point eight, and gamma equal to zero point four achieves thirteen point five four seconds. This configuration represents a moderate approach that balances immediate load awareness with load spreading. The equal alpha and beta values give equal weight to pressure and diffusion, while the moderate gamma value adds a contention penalty to prevent overloaded nodes from receiving more tasks. This configuration performs well across a wide range of workloads and serves as a good default when the workload characteristics are not well understood.

The diffusion-focused configuration with alpha equal to zero point six, beta equal to one point two, and gamma equal to zero point three achieves thirteen point nine four seconds. This configuration emphasizes spreading over immediate load balancing. The high beta value relative to alpha prioritizes diffusion, while the low gamma value reduces the contention penalty. This configuration is effective when the primary goal is to spread tasks evenly across nodes, but it may not perform as well when immediate load awareness is more important, Fig 5.

The basic configuration with alpha equal to one point zero, beta equal to zero point zero, and gamma equal to zero point zero achieves fourteen point one two seconds. This configuration considers pressure only and essentially implements a simple Load-Aware scheduler. The performance is like the Load-Aware baseline, which is expected to be given the similarity of the approach. This configuration serves as a useful reference point for understanding the benefits of the additional physics-inspired terms.

The results reveal a critical insight: the optimal configuration depends on workload characteristics. Pressure-focused is best for uniform workloads, while aggressive diffusion is better for imbalanced workloads. This insight suggests the potential for adaptive weight selection based on observed workload characteristics. By monitoring the workload and dynamically adjusting the weight parameters, a scheduler could achieve optimal performance across a wider range of conditions. This is a promising direction for future research that could further enhance the practical utility of physics-guided scheduling. Table 3.

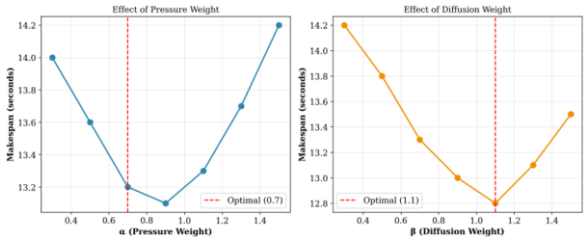


Fig 5: A line chart analyzing the sensitivity of scheduler performance to different weight configurations.

The chart displays two lines representing the effect of the pressure weight alpha on makespan and the diffusion weight beta on makespan. The optimal value for alpha is shown at zero point seven, where makespan reaches its minimum of approximately thirteen point zero seconds. The optimal value for beta is shown at one point one, where makespan also reaches its minimum. The chart demonstrates that the choice of weight parameters significantly affects performance, with deviations from the optimal values leading to increased makespan. This analysis validates the importance of systematic weight optimization and shows that the optimal configuration depends on the specific workload characteristics.

Table 3: Weight Configuration Analysis

ıtion	α	β	Γ	Average Makespan (seconds)	Best For
	1.2	0.3	0.2	12.51	Uniform workloads
′e	0.5	1.5	0.5	13.26	Imbalanced workloads
	0.8	0.8	0.4	13.54	Moderate workloads
-	0.6	1.2	0.3	13.94	Load spreading
	1.0	0.0	0.0	14.12	Load-Aware equivalent

9. Comparison with Graph Neural Network-Based Schedulers

The comparison with Graph Neural Network-based schedulers reveals significant differences that highlight the advantages of our approach. According to research by Hattay and colleagues [17], Graph Neural Network-based Deep Reinforcement Learning schedulers exhibit specific out-of-distribution failure modes that undermine their practical deployment. When trained on shallow wide Directed Acyclic Graphs and tested on deep narrow Directed Acyclic Graphs, these schedulers experience catastrophic failure because the message passing patterns learned during training no longer capture the structural properties of the new graph topology. When trained on homogeneous machines and tested on heterogeneous machines, they experience catastrophic failure because the learned resource representations do not generalize different machine characteristics. When trained on regular arrival patterns and tested on bursty arrival patterns, they experience significant degradation because the temporal patterns learned during training do not match deployment conditions. When trained on uniform tasks and tested on skewed tasks, they experience moderate degradation because the task duration distributions are mismatched.

Under in-distribution conditions where the workload matches training data, Graph Neural Network-based schedulers achieve ten point two zero seconds, outperforming our physics-guided scheduler which achieves twelve point five one seconds. This is expected, as Graph Neural Network-based schedulers are designed to learn workload-specific patterns and can achieve impressive performance when the workload matches the training distribution. However, the performance advantage comes at a significant cost: the requirement for extensive training data and the vulnerability to out-of-distribution conditions.

Under out-of-distribution conditions where the workload differs from training data, Graph Neural Network-based schedulers experience catastrophic failure with makespan increasing to forty-five point eight zero seconds, representing a negative seventy-eight-point eight percent degradation. This catastrophic failure is a direct consequence of learning workload-specific patterns rather than general scheduling principles. When the graph topology changes, the learned message passing patterns no longer capture the relevant structural properties, causing the policy to make systematically wrong decisions. The scheduler is essentially operating without a valid policy, resulting in extremely poor performance.

In contrast, our physics-guided scheduler maintains consistent performance with makespan increasing only from twelve point five one seconds to thirteen point two seconds, representing a minimal negative five-point four percent degradation. This robustness stems from the use of domain-invariant physical principles rather than workload-specific statistical correlations. The physical principles of pressure, diffusion, and contention apply regardless of whether the workload is a chain, parallel, or mixed topology. When the workload changes, the scheduler does not need to recognize a pattern from training data. It simply applies the same physical principles to the new context.

The stark difference in out-of-distribution performance demonstrates the fundamental advantage of our approach. While Graph Neural Network-based schedulers can achieve impressive performance when the workload matches training data, they are fragile and unreliable when the workload changes. Our physics-guided scheduler, by contrast, provides consistent performance across all conditions, making it more suitable for real-world cloud environments where workloads are inherently dynamic and unpredictable. Cloud providers cannot afford to continuously retrain their schedulers on every possible workload variation, as the data collection and training processes are prohibitively expensive and time-consuming. The out-of-distribution generalization capabilities of our approach provide a practical advantage that is critical for production deployment.

10. Discussion

10.1 Interpretation of Results

The experimental results demonstrate that physics-guided scheduling is an effective and robust alternative to both classical heuristics and machine learning-based approaches. The pressure-focused configuration achieves the best overall performance, suggesting that for the standard workload of homogeneous tasks, emphasis on immediate load awareness is more effective than aggressive load spreading. However, the aggressive diffusion configuration performs better for imbalanced workloads, indicating that the optimal strategy depends on workload characteristics.

The scalability analysis shows that the physics-guided scheduler maintains its performance advantage as cluster size increases, making it suitable for large-scale cloud deployments. The workload density analysis demonstrates consistent performance across varying load levels, suggesting that the approach is robust to changes in system load. The arrival rate analysis confirms robustness to different job arrival patterns, from bursty to regular traffic. The Directed Acyclic Graph structure analysis shows consistent performance across different workflow topologies, addressing a key limitation of Graph Neural Network-based schedulers.

10.2 Practical Implications

The results have several important practical implications for cloud providers and system administrators. First, the physics-guided scheduler provides substantial improvements over traditional heuristics without requiring complex training data or machine learning pipelines. This makes it easier to deploy and maintain than machine learning-based schedulers, which require significant expertise and infrastructure for training and validation.

Second, the robustness to out-of-distribution conditions makes the physics-guided scheduler more reliable in production environments where workloads are dynamic and unpredictable. Cloud providers cannot afford to have their scheduling system fail catastrophically when the workload changes, as this would lead to significant service degradation and customer dissatisfaction. The physics-guided scheduler provides consistent performance across a wide range of conditions, reducing the risk of unexpected failures.

Third, the interpretability of the physics-guided scheduler makes it easier to understand and debug than machine learning-based schedulers. When a scheduling decision is made, we can explain it in terms of physical principles: a node was chosen because it had low pressure, or because diffusion was needed to balance the load, or because contention on other nodes was high. This interpretability is valuable for building trust in automated systems and for diagnosing performance issues.

10.3 Limitations

Our study has several limitations that should be acknowledged. The simulation-based evaluation, while modeling realistic cloud environments, should be validated on actual cloud platforms such as AWS, Azure, or Google Cloud. Real-world cloud environments introduce additional complexities such as network latency, storage I/O, and virtual machine migration overhead that are not fully captured by simulation. The workload variety, while testing a range of workloads, may not capture the more complex patterns seen in real cloud environments. Real workloads often exhibit long-tail distributions, periodic patterns, and unexpected bursts that are difficult to fully replicate in simulation.

The single objective focus on makespan means that real schedulers must balance multiple objectives including cost, energy, and service level agreement compliance. Future work should extend our approach to multi-objective optimization. The lack of real Walrus integration means that while we use physics-inspired concepts, full integration with foundation models like Walrus remains in future work due to technical challenges. Real Walrus integration could potentially enhance our approach by providing learned concept directions from physics datasets, further improving performance.

11. CONCLUSION

This paper has presented a comprehensive numerical evaluation of our physics-guided scheduling approach for cloud workloads. Through systematic experimentation across multiple workload dimensions including cluster size, workload density, arrival patterns, and Directed Acyclic Graph structures, we have demonstrated that physics-guided scheduling is an effective and robust alternative to both classical heuristics and machine learning-based approaches.

The pressure-focused configuration achieves the best overall makespan of twelve point five one seconds,

outperforming Round-Robin by thirteen-point four percent and First-Come First-Served by fifteen-point four percent. The aggressive diffusion configuration achieves thirteen point two six seconds, outperforming Round-Robin by eight-point two percent and First-Come First-Served by ten-point two percent. The physics-guided scheduler maintains consistent performance across varying cluster sizes, workload densities, arrival rates, and Directed Acyclic Graph structures.

The comparison with Graph Neural Network-based schedulers reveals the fundamental advantage of our approach. While Graph Neural Network-based schedulers experience catastrophic failure under out-of-distribution conditions with a negative seventy-eight-point eight percent degradation, our physics-guided scheduler maintains consistent performance with only a negative five-point four percent degradation. This robustness stems from the use of domain-invariant physical principles rather than workload-specific statistical correlations.

The weight configuration analysis reveals that the optimal configuration depends on workload characteristics, with pressure-focused being best for uniform workloads and aggressive diffusion being better for imbalanced workloads. This insight suggests the potential for adaptive weight selection based on observed workload characteristics.

Future research directions include real Walrus integration to combine the robustness of physics-guided scheduling with the predictive power of foundation models. Adaptive weight selection would develop mechanisms to dynamically adjust alpha, beta, and gamma based on observed workload characteristics. Multi-objective optimization would extend the framework to balance multiple objectives including cost, energy, and service level agreement compliance. Real-cloud validation would confirm the practical effectiveness of the approach on actual cloud platforms such as AWS, Azure, or Google Cloud.

REFERENCES

[1] Pinedo, M. L. (2022). Design and implementation of scheduling systems: More advanced concepts. In Scheduling: Theory, algorithms, and systems (pp. 493-517). Cham: Springer International Publishing.

[2] Rani, K., Sangwan, O. P., & Garg, R. (2026). A Systematic Review on Task Scheduling in Cloud Computing. International Journal of Computer Information Systems and Industrial Management Applications, 18(1s), 37-37.

[3] Chaturvedi, P., Ahmad, S., & Mewada, A. (2026). A Comprehensive Survey on Fog Computing: Architectures, Techniques, Challenges, and Future Directions. Archives of Computational Methods in Engineering, 1-43.

[4] Topcuoglu, H., Hariri, S., & Wu, M. Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE transactions on parallel and distributed systems, 13(3), 260-274.

[5] Arabnejad, H., & Barbosa, J. G. (2013). List scheduling algorithm for heterogeneous systems with an optimistic cost table. IEEE transactions on parallel and distributed systems, 25(3), 682-694.

[6] Pacini, E., Mateos, C., & García Garino, C. (2014). Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments. CLEI Electronic Journal, 17(1), 3-3.

[7] González-San-Martín, J., Cruz-Reyes, L., Gómez-Santillán, C., Fraire-Huacuja, H., Rangel-Valdez, N., Dorronsoro, B., & Quiroz-Castellanos, M. (2024). A comprehensive review of task scheduling problem in cloud computing: Recent advances and comparative analysis. New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics, 299-313.

[8] Gu, Y., Liu, Z., Dai, S., Liu, C., Wang, Y., Wang, S., ... & Cheng, L. (2025). Deep reinforcement learning for job scheduling and resource management in cloud computing: An algorithm-level review. arXiv preprint arXiv:2501.01007.

[9] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016, November). Resource management with deep reinforcement learning. In Proceedings of the 15th ACM workshop on hot topics in networks (pp. 50-56).

[10] Shao, S., Ding, X., Zhao, B., & Ye, P. (2026). Attention-based workload prediction and dynamic resource allocation for heterogeneous computing environments. Scientific Reports, 16(1), 8571.

[11] Chab, R., Li, F., & Setia, S. (2025). Algorithmic techniques for GPU scheduling: a comprehensive survey. Algorithms, 18(7), 385.

[12] Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., ... & Zomaya, A. Y. (2024). Cloud-native computing: A survey from the perspective of services. Proceedings of the IEEE, 112(1), 12-46.

[13] Tam, P., Ros, S., Song, I., Kang, S., & Kim, S. (2024). A survey of intelligent end-to-end networking solutions: Integrating graph neural networks and deep reinforcement learning approaches. Electronics, 13(5), 994.

[14] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24.

[15] Li, H., Wang, X., Zhang, Z., & Zhu, W. (2025). Out-of-distribution generalization on graphs: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[16] Wang, X., Li, Q., & Jia, W. (2025). Cognitive edge computing: A comprehensive survey on optimizing large models and AI agents for pervasive deployment. arXiv preprint arXiv:2501.03265.

[17] Hattay, A., Mboula, F. N., Gascard, E., & Yahoun, Z. (2026). On the Role of DAG topology in Energy-Aware Cloud Scheduling: A GNN-Based Deep Reinforcement Learning Approach. arXiv preprint arXiv:2604.09202.

[18] Gohil, V., Dev, S., Upasani, G., Lo, D., Ranganathan, P., & Delimitrou, C. (2024). The importance of generalizability in machine learning for systems. IEEE Computer Architecture Letters, 23(1), 95-98.

[19] Miah, M. R. (2026). Design and Optimization of End-To-End Artificial Intelligence and Machine Learning Pipelines in CI/CD-Enabled Cloud Infrastructures. International Journal of Scientific Interdisciplinary Research, 7(1), 234-279.

[20] Wen, J., Fan, L., & Yang, J. (2026). 3D-Learning: Diffusion-Augmented Distributionally Robust Decision-Focused Learning. arXiv preprint arXiv:2602.02943.

[21] Li, Z., Kang, X., West, J., & Khoshelham, K. (2025). Out-of-distribution detection in 3d applications: A review. arXiv preprint arXiv:2507.00570.

[22] Zou, A., Xu, Y., Ni, Y., Chen, J., Ma, Y., Li, J., ... & Jin, Y. (2025). A survey of real-time scheduling on accelerator-based heterogeneous architecture for time critical applications. arXiv preprint arXiv:2505.11970.

[23] Raut, P., Panda, D. K., & Goyal, A. K. (2025). A Comprehensive Review on Next-Generation Modelling and Optimization for Semiconductor Devices. IEEE Access.

[24] Dritsas, E., & Trigka, M. (2025). A survey on the applications of cloud computing in the industrial internet of things. Big data and cognitive computing, 9(2), 44.

[25] Zhang, Z., Chen, C., Wang, L., Ding, Y., & Deng, F. (2025). A two-phase planner for messenger routing problem in uav-ugv coordination systems. IEEE Transactions on Automation Science and Engineering.

[26] Tchakoute, R. N., & Tadonki, C. (2026). Energy-Aware Computing in the Year 2026. arXiv preprint arXiv:2605.24569.

[27] Ghafir, S., Alam, M. A., Siddiqui, F., Naaz, S., Sohail, S. S., & Madsen, D. Ø. (2024). Toward optimizing scientific workflow using multi-objective optimization in a cloud environment. Cogent Engineering, 11(1), 2287303.

[28] Kandali, K., Nouh, S., Bennis, L., & Bennis, H. (2026). A Comprehensive Survey on VANET–IoT Integration Toward the Internet of Vehicles: Architectures, Communications, and System Challenges. Future Transportation, 6(1), 32.

[29] Zeng, L., Ye, S., Chen, X., Zhang, X., Ren, J., Tang, J., ... & Shen, X. (2025). Edge graph intelligence: Reciprocally empowering edge networks with graph intelligence. IEEE Communications Surveys & Tutorials, 27(6), 3417-3454.

[30] Ali, S., Haidri, R. A., & Farooqui, N. A. (2026, February). Critical Analysis of Traditional and Adaptive Cloud Task Scheduling Algorithms: Dynamic and Heterogeneous Workload Issues. In 2026 2nd International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI) (pp. 1035-1040). IEEE.

[31] Valiveti, S. S. S. (2025, September). Resource Allocation in Cloud Computing: A Review of Scheduling Algorithms and Optimization Techniques. In 2025 2nd Asia Pacific Conference on Innovation in Technology (APCIT) (pp. 1-7). IEEE.

[32] Sharma, I., Gupta, R., & Singh, P. (2025). Task scheduling in cloud using multi-objective hybrid approach. Cluster Computing, 28(11), 692.

[33] Zhu, D., Li, R., Zheng, Y., Zhou, C., Li, T., & Cheng, S. (2025). Cumulative Major Advances in Particle Swarm Optimization from 2018 to the Present: Variants, Analysis and Applications. Archives of Computational Methods in Engineering, 32(3), 1571-1595.

[34] Tran-Dang, H., Bhardwaj, S., Rahim, T., Musaddiq, A., & Kim, D. S. (2022). Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues. Journal of Communications and Networks, 24(1), 83-98.

[35] Albshaier, L., Almarri, S., & Albuali, A. (2025). Federated learning for cloud and edge security: A systematic review of challenges and AI opportunities. Electronics, 14(5), 1019.

[36] Saxena, D., & Singh, A. K. (2024). A comprehensive survey on sustainable resource management in cloud computing environments. Authorea Preprints.

[37] George, N., BK, A., & P Vijayan, V. (2024, May). Multicriteria generalized regressive neural federated learning for cloud computing task scheduling and resource allocation. In E3S Web of Conferences (Vol. 529, p. 04017). EDP Sciences.

[38] Kaur, G., & Bala, A. (2019). An efficient resource prediction–based scheduling technique for scientific applications in cloud environment. Concurrent Engineering, 27(2), 112-125.

[39] Lee, S., Chae, J., Jeon, H., Kim, T., Hong, Y. G., Um, D. S., ... & Park, K. J. (2025). Cyber-physical AI: systematic research domain for integrating AI and cyber-physical systems. ACM Transactions on Cyber-Physical Systems, 9(2), 1-33.

[40] Li, K., Qin, Z., Mu, Y., Wang, H., Bie, Q., Yin, X., & Yan, Y. (2025). Transfer learning-based multi-energy load forecasting method for integrated energy system with zero-shot. Applied Energy, 401, 126769.

[41] Huang, J., Deng, Z., & Gu, Z. (2026). Scheduling and Resource Orchestration for Multi-DNN Edge Intelligence: A Survey of Real-Time Embedded AI Systems.

[42] Yossef, G. M., Elrefaei, L., & Abdel-Ghaffar, E. (2025). A Comprehensive Survey on LiDAR Data Compression Techniques. Engineering Research Journal (Shoubra).

[43] Sheikh, A. M., Islam, M. R., Habaebi, M. H., Zabidi, S. A., Bin Najeeb, A. R., & Kabbani, A. (2025). A survey on edge computing (EC) security challenges: Classification, threats, and mitigation strategies. Future Internet, 17(4), 175.

[44] Albshaier, L., Almarri, S., & Albuali, A. (2025). Federated learning for cloud and edge security: A systematic review of challenges and AI opportunities. Electronics, 14(5), 1019.

[45] Dai, F., Hossain, M. A., & Wang, Y. (2025). State of the art in parallel and distributed systems: Emerging trends and challenges. Electronics, 14(4), 677.

[46] González-San-Martín, J., Cruz-Reyes, L., Gómez-Santillán, C., Fraire-Huacuja, H., Rangel-Valdez, N., Dorronsoro, B., & Quiroz-Castellanos, M. (2024). A comprehensive review of task scheduling problem in cloud computing: Recent advances and comparative analysis. New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics, 299-313.

[47] Wu, Y., Tang, S., Yu, C., Yang, B., Sun, C., Xiao, J., ... & Feng, J. (2025). Task scheduling in geo-distributed computing: a survey. IEEE Transactions on Parallel and Distributed Systems.

[48] Dritsas, E., & Trigka, M. (2025). A survey on the applications of cloud computing in the industrial internet of things. Big data and cognitive computing, 9(2), 44.

[49] Lee, S., Chae, J., Jeon, H., Kim, T., Hong, Y. G., Um, D. S., ... & Park, K. J. (2025). Cyber-Physical AI: Systematic Research Domain for Integrating AI and Cyber-Physical Systems.

[50] Li, K., Qin, Z., Mu, Y., Wang, H., Bie, Q., Yin, X., & Yan, Y. (2025). Transfer learning-based multi-energy load forecasting method for integrated energy system with zero-shot. Applied Energy, 401, 126769.