MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of Higher Education
Peter the Great St. Petersburg Polytechnic University

Institute of Computer Science and Cybersecurity

Higher School of Artificial Intelligence Technology

Direction 02.03.01 Mathematics and computer Science

Literature Review

Workflow Scheduling Algorithms for High-Performance Computing and Cloud Environments

Student,
group 5130201/20102 _____ Gaar V. S.

Supervisor, Ph. D. _____ Motorin D. E.

«_____»_____ 2024г.

Saint-Petersburg, 2024

Contents

Abstract

Workflow scheduling is critical in optimizing the performance of high-performance computing (HPC) and cloud environments, where efficient resource allocation and task execution are paramount. This literature review provides a comprehensive comparative analysis of ten advanced workflow scheduling algorithms developed for various computing environments, including HPC, mobile edge computing (MEC), cloud computing, and hybrid systems. The methodologies are examined concerning their mathematical foundations, optimization techniques, application domains, and adaptability to dynamic workloads and system scalability. Performance outcomes are analyzed based on execution time optimization, energy efficiency, cost reduction, privacy preservation, and resource utilization. The review identifies common strengths such as adaptability and multi-objective optimization, while also highlighting key differences and trade-offs in addressing conflicting objectives. Limitations are discussed, and future research directions are proposed, including the integration of renewable energy sources, enhanced privacy mechanisms, and the exploration of hybrid optimization techniques. The findings underscore the significant role of advanced scheduling algorithms in improving system performance and resource management in HPC and cloud environments.

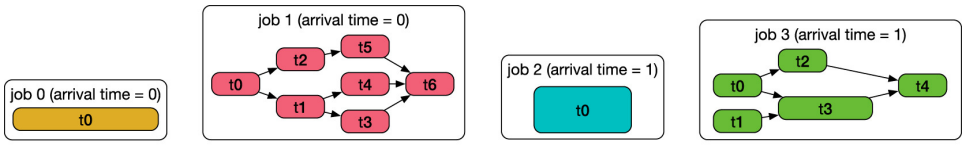
Keywords

Workflow Scheduling, High-Performance Computing (HPC), Mobile Edge Computing (MEC), Cloud Computing, Directed Acyclic Graph (DAG), Resource Allocation, Data Privacy, Optimization Algorithms, Energy Efficiency, Latency Reduction.

1 Introduction

Workflow scheduling plays a pivotal role in modern computing environments, where the complexity and scale of tasks have grown exponentially. These environments include High-Performance Computing (HPC) [1], Mobile Edge Computing (MEC) [2], [3], Edge Function as a Service (Edge FaaS) [4], Geographically Distributed Cloud Data Centers (GD-CDCs) [5], [6], and hybrid cloud-edge ecosystems [7], [8], each characterized by unique challenges and requirements. The surge in data-intensive applications – ranging from artificial intelligence workloads to real-time IoT systems – has amplified the demand for efficient scheduling mechanisms that can dynamically allocate resources, optimize execution time, and minimize energy consumption while adhering to evolving privacy regulations.

The advent of distributed systems introduces complexities that traditional scheduling methods struggle to address. Distributed computing environments are inherently heterogeneous, involving diverse resource types such as CPUs, GPUs, memory, and network bandwidth, which must be coordinated to meet the demands of complex workflows. A common representation of such workflows is through a Directed Acyclic Graph (DAG), as shown in Fig. 1 of [1], where nodes represent tasks, and edges define dependencies. Moreover, the dynamic nature of workloads, where task priorities and resource availability fluctuate unpredictably, exacerbates the need for adaptable and resilient scheduling mechanisms [9]. Additionally, these systems often span multiple geographic regions, as seen in GD-CDCs and hybrid cloud models, where data transfer costs, latency, and privacy constraints further complicate resource allocation [6], [8].



(a) 1 task. (b) 7 tasks. (c) 1 task. (d) 5 tasks.

Fig 1. Examples of tasks and jobs with distinct arrival times and demands (in terms of processing and expected walltime). Each rectangle represents a task: processing is denoted by the rectangle height and execution time by the base's length. The arrows represent the execution's workflow.

Energy efficiency has emerged as a critical consideration, particularly in edge and cloud computing environments, where power consumption directly impacts operational costs and sustainability goals. Studies such as exploration of DVFS-based optimization [5] highlight the growing emphasis on energy-aware scheduling, which seeks to balance resource utilization with energy savings. Similarly, edge-focused systems like those addressed in [9] aim to optimize energy consumption while ensuring low latency, which is critical for real-time applications such as autonomous vehicles and industrial IoT.

Another dimension of complexity lies in the integration of data privacy requirements, particularly in geo-distributed systems. As workflows increasingly operate across multiple jurisdictions, privacy-preserving mechanisms, such as those proposed in [6], are essential to comply with regional data protection laws, reflected in the Fig. 2 of [6] while maintaining operational efficiency. These privacy constraints introduce new trade-offs between WAN usage, data locality, and computational overhead, necessitating innovative approaches to

graph partitioning and workflow scheduling [7].

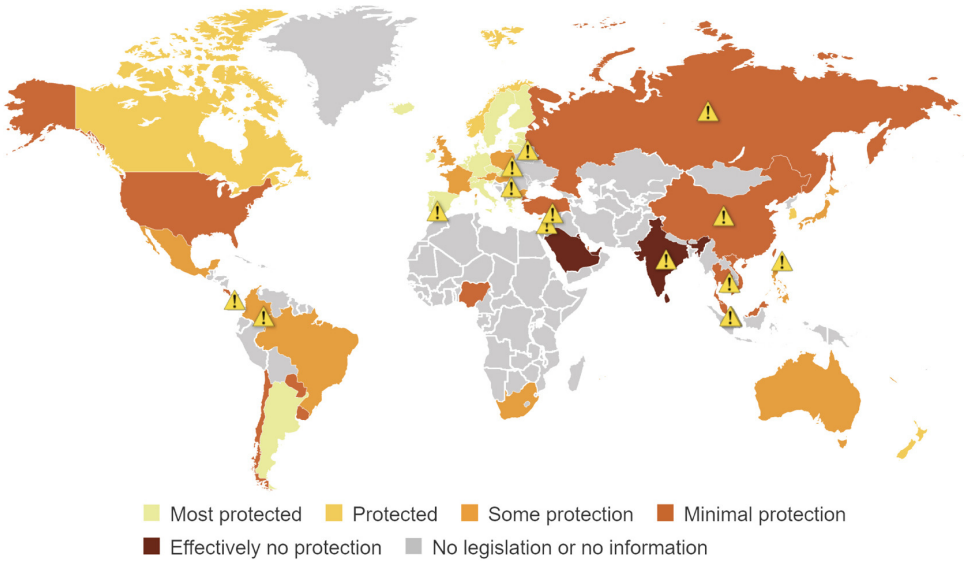


Fig 2. A comparative ranking of 54 countries' privacy and data protection requirements.

The core challenges shared across all ten reviewed studies include:

- Efficient resource allocation in heterogeneous environments, ensuring fair distribution of computational resources such as CPU, memory, and network bandwidth.
- Dynamic task scheduling to adapt to unpredictable workloads and fluctuating resource availability in real time.
- Optimization of key performance metrics, such as execution time, cost, energy consumption, and latency, to achieve global system efficiency.
- Scalability and flexibility, enabling systems to handle growing workloads and infrastructure expansions without significant performance degradation.
- Integration of privacy and regulatory constraints, particularly in systems spanning multiple regions with varying legal requirements [6], [10].

To address these challenges, researchers have turned to machine learning, heuristic and metaheuristic optimization algorithms, and hybrid approaches that combine static and dynamic scheduling strategies. Examples include the use of Actor-Critic Reinforcement Learning [1], multi-strategy heuristic optimization [9], and genetically modified particle swarm optimization [10]. These methods represent a shift towards more adaptable, intelligent scheduling frameworks that balance multiple objectives simultaneously.

This literature review focuses on the comparative analysis of these innovative scheduling approaches, highlighting their strengths, limitations, and potential for integration into future systems. The studies collectively advance the field by addressing the pressing need for scalable, energy-efficient, and privacy-compliant workflow scheduling, offering insights into how these methods can be adapted for the evolving landscape of distributed computing.

The review is structured as follows: Section 2 provides background and context, discussing the significance of workflow scheduling, the challenges faced, and the evolution

from traditional to advanced algorithms. Section 3 presents a comparative analysis of the scheduling methods, including a detailed examination of their mathematical and programming approaches, optimization techniques, application domains, and adaptability. Section 4 analyzes and compares the performance outcomes of the algorithms based on key metrics. Section 5 synthesizes the findings, highlighting common strengths, key differences, and practical implications. Section 6 proposes future research directions based on identified gaps and limitations. Finally, Section 7 concludes the review by summarizing the main findings and emphasizing the contributions of advanced scheduling algorithms to HPC and cloud environments.

2 Background and Context

2.1 Workflow Scheduling in HPC and Cloud Environments

Workflow scheduling is the process of mapping and managing the execution of interdependent computational tasks across available resources to optimize specific performance criteria. In HPC and cloud environments, workflows often consist of complex applications that require significant computational power and efficient resource management. The representation of workflows as Directed Acyclic Graphs (DAGs) allows for a structured depiction of tasks and their dependencies, facilitating the development of scheduling strategies that can handle complex interrelations among tasks.

In HPC environments, workflow scheduling aims to maximize resource utilization and minimize execution times for applications such as scientific simulations, data analysis, and machine learning tasks. Cloud environments introduce additional considerations, including resource elasticity, virtualization, and cost models based on resource usage. The integration of edge computing and MEC further complicates scheduling due to the need for low-latency processing and the limitations of edge devices in terms of computational capacity and energy resources.

2.2 Challenges in Workflow Scheduling

Workflow scheduling faces several challenges in modern computing environments:

- Heterogeneous Resources: Computing environments comprise a diverse range of resources with varying capabilities, such as CPUs, GPUs, memory capacities, and network bandwidths. Scheduling algorithms must account for this heterogeneity to optimize performance and resource utilization.

- Dynamic Workloads: Workloads can fluctuate unpredictably due to varying user demands and application requirements. Scheduling strategies need to be adaptive to handle real-time changes in workload intensity and resource availability.

- Data Privacy and Compliance: The distribution of data across different geographic regions introduces challenges related to data privacy regulations, such as the General Data Protection Regulation (GDPR). Scheduling algorithms must ensure compliance by incorporating data locality and privacy-preserving mechanisms.

- Energy Efficiency and Cost Optimization: Reducing energy consumption is crucial for both environmental sustainability and operational cost savings. Scheduling methods must balance performance objectives with energy efficiency considerations.

- Latency and Quality of Service (QoS): Applications in MEC and IoT environments often have strict latency requirements. Scheduling algorithms must minimize delays to meet QoS demands while managing limited computational resources.

2.3 Traditional vs. Advanced Scheduling Algorithms

Traditional scheduling algorithms, such as First-Come, First-Served (FCFS) and Shortest Processing Time (SPT), are generally static and heuristic-based. While they offer simplicity and ease of implementation, they lack the flexibility to adapt to dynamic and heterogeneous computing environments. These methods often fail to consider the complex dependencies and resource constraints inherent in modern workflows, leading to suboptimal performance and resource utilization.

Advanced scheduling algorithms have emerged to address these limitations. By leveraging machine learning techniques, metaheuristic optimization, and hybrid approaches, these methods provide enhanced adaptability and efficiency. For instance, reinforcement learning allows scheduling policies to be learned and optimized based on continuous feedback from the environment. Metaheuristic algorithms, such as genetic algorithms and particle swarm optimization, enable the exploration of large solution spaces to find near-optimal scheduling configurations. Hybrid approaches combine multiple optimization strategies to balance global exploration and local exploitation effectively.

These advanced algorithms are better equipped to handle the complexities of modern computing environments, offering the potential to improve performance metrics significantly while accommodating the dynamic nature of workloads and resource availability.

3 Comparative Analysis of Scheduling Methods

3.1 Overview of Scheduling Methods

The ten selected research articles present a diverse range of scheduling methods tailored to specific computing environments and objectives:

1. [1]: Utilizes Actor-Critic Reinforcement Learning (ACRL) for scheduling DAG-based workflows in HPC data centers, aiming to minimize task delays and optimize resource utilization.

2. [2]: Introduces the Feedback Artificial Remora Optimization (FARO) algorithm for secure workflow scheduling in MEC environments, focusing on reducing CPU and memory utilization while enhancing security.

3. [3]: Proposes an opposition-based Marine Predator Algorithm (OMPA) combined with workload prediction using artificial neural networks for multi-workflow scheduling in MEC environments.

4. [4]: Develops an assignment mechanism inspired by the course allocation problem for scheduling workflows in Edge Function as a Service (EFaaS) environments, emphasizing efficient resource allocation and reduced execution times.

5. [5]: Presents the Electricity Price and Energy-Efficient (EPEE) scheduling framework for geographically distributed cloud data centers, incorporating dynamic electricity pricing and energy consumption optimization.

6. [6]: Introduces the Privacy-Preserving Partitioning-based Scheduling Algorithm (PPPS) for geo-distributed data centers, addressing privacy constraints and network heterogeneity.

7. [7]: Presents a multi-resource scheduling algorithm for moldable workflows in HPC systems, focusing on maximizing resource utilization and minimizing workflow execution times.

8. [8]: Proposes Hybrid Scheduling for Hybrid Clouds (HSHC), combining genetic algorithms with dynamic adjustments to optimize the scheduling of scientific workflows in hybrid cloud environments.

9. [9]: Proposes the Multi-Strategy Improved Sand Cat Optimization Algorithm (MSISCSOA) for workflow scheduling in heterogeneous edge computing environments, targeting the minimization of execution latency and energy consumption.

10. [10]: Develops a Genetically-Modified Multi-objective Particle Swarm Optimization (GM-MOPSO) approach for HPC workflow scheduling, aiming to balance execution time and energy efficiency.

3.2 Detailed Comparative Analysis

An Actor-Critic RL approach was used in [1], designed to reduce task delays in high-performance computing environments. By framing task scheduling as a directed acyclic graph (DAG), this method allows adaptive queue management, significantly outperforming traditional First-Come, First-Served (FCFS) and Shortest Processing Time (SPT) methods, particularly in terms of throughput. However, its focus on HPC makes it less suited for environments with dynamic workloads or stringent privacy constraints.

In comparison, Feedback Artificial Remora Optimization (FARO) was applied in [2] for MEC environments, balancing CPU, memory, and security requirements. FARO shows high efficiency, achieving low CPU and memory usage (0.012 and 0.010, respectively), which is especially beneficial for mobile edge scenarios. Unlike the method in [1], FARO highlights resource security as a core feature, providing a hybrid optimization approach for security-sensitive workflows where resource availability varies dynamically.

The Multi-Strategy Improved Sand Cat Optimization Algorithm (MSISCSOA), introduced in [9], focuses on reducing delay and energy consumption in heterogeneous edge computing environments. With energy consumption decreased by approximately 19.56%, MSISCSOA emphasizes task adaptability through dynamic search strategies. This method contrasts with FARO's emphasis on security [2], instead optimizing energy efficiency and latency in resource-constrained edge networks. The method's edge-specific design yields lower delay compared to general algorithms, aligning well with latency-critical applications.

In EFaaS environments, a serverless scheduling mechanism combining Highest Bid First and Warm Function First (HBFM and WFFM) is proposed in [4] to enhance execution time. The mechanism is unique in its prioritization and bidding-based resource allocation, achieving efficient multi-user task distribution. Results indicate a decrease in workflow execution time, but unlike the MSISCSOA approach [9], this method is less effective for energy optimization, as it prioritizes task execution time over energy or CPU considerations in serverless environments.

Electricity Price and Energy-Efficient (EPEE) Scheduling, presented in [5], is particularly notable for its energy cost reduction in geographically distributed data centers, where energy consumption varies by location. Leveraging Dynamic Voltage and Frequency Scaling (DVFS), EPEE significantly cuts down energy costs, aligning with the efficiency goals of MSISCSOA [9] but extending them to cloud environments with geographic data distribution. Unlike edge-specific approaches, EPEE accommodates varying electricity prices, making it ideal for distributed cloud systems where energy cost control is crucial.

The method in [3] further innovates within MEC by combining a Marine Predator Algorithm (OMPA) with workload prediction via artificial neural networks (ANNs). The OMPA's unique opposition-based learning prevents local minima, optimizing task scheduling even under fluctuating workloads. Compared to the MSISCSOA algorithm [9], which also targets energy and delay, the method presented in [3] offers superior deadline compliance and VM usage reduction by dynamically predicting workloads, addressing unpredictability in MEC with high efficiency.

Furthermore, [6] focuses on privacy in geo-distributed data centers through Privacy-Preserving Partitioning-based Scheduling (PPPS). Their results show remarkable reductions in WAN usage (up to 99%) and execution time (up to 93%) by addressing complex multi-level privacy constraints. Unlike other studies focused primarily on resource and energy efficiency [1], [3]–[5], [7]–[10], PPPS uniquely addresses regulatory requirements in data privacy. This feature sets it apart as a solution for environments where data transfer is restricted by privacy laws, particularly in scientific workflows operating across multiple jurisdictions.

The Multi-Resource Scheduling Algorithm (MRSA), introduced in [7], focuses on optimizing moldable workflows in HPC systems. By allowing resource reallocation before task execution, MRSA provides a high degree of flexibility, making it particularly effective in managing heterogeneous resources like CPU, memory, and I/O. Unlike Actor-Critic RL used in [1] or FARO in [2], MRSA employs a heuristic-based optimization strategy tailored to HPC environments. This emphasis on multi-resource scheduling aligns conceptually with the multi-strategy optimization in [9] but extends it to include HPC-specific considerations.

Hybrid Scheduling for Hybrid Clouds (HSHC), presented in [8], combines static genetic algorithms with dynamic adjustments to handle hybrid cloud workflows effectively. HSHC is particularly notable for its data locality optimization, which minimizes data transfer costs across cloud environments. This method shares similarities with EPEE [5], which also targets geo-distributed systems but focuses on energy consumption rather than data locality. Additionally, the two-phase structure of HSHC resembles PPPS [6], where distinct optimization stages address different aspects of workflow scheduling.

The Genetically-Modified Multi-Objective Particle Swarm Optimization (GMPSO) algorithm, proposed in [10], integrates genetic operations into PSO for optimizing cost and makespan in hybrid cloud systems. This approach mirrors multi-objective optimization strategies seen in MSISCSOA [9] and FARO [2] but differentiates itself with its unique matrix coding of tasks and resources, offering more granular control over workflow execution. Compared to the heuristic-based MRSA [7] or dynamic HSHC [8], GMPSO is better suited for scenarios requiring simultaneous optimization of multiple objectives.

4 Comparative Analysis of Results

4.1 Overview of Performance Metrics

The performance metrics used to evaluate the scheduling algorithms include:

- Execution Time (Makespan) Optimization: The total time required to execute the entire workflow.
- Energy Efficiency: The amount of energy consumed during task execution, with an emphasis on reducing overall energy usage.
- Cost Optimization: Operational costs associated with resource usage and energy consumption.
- Privacy Preservation and Data Locality: Compliance with data privacy regulations and optimization of data placement to reduce data transfer costs and latency.
- Resource Utilization and Scalability: Effective use of available resources and the ability to scale with increasing workloads.

4.2 Detailed Comparative Analysis

The ten reviewed studies exhibit significant advancements in workflow scheduling, each tailored to address specific challenges within diverse computing environments. This section analyzes the similarities and differences in their results, highlighting shared achievements, unique strengths, and limitations.

A primary focus across the studies is the optimization of execution time, though each employs distinct methods tailored to their specific environments.

- Algorithm in the [1] achieve a 35% improvement in DAG processing speed, demonstrating the effectiveness of Actor-Critic RL in prioritizing task dependencies in HPC environments. This result parallels the performance of MRSA in [7], which also focuses on HPC workflows but achieves optimization by dynamically reallocating resources before execution.
- Similarly, HSHC in [8] reduces execution time by 25% in hybrid cloud environments through a combination of genetic algorithms and dynamic scheduling, while GMPSO in [10] balances execution time with cost, offering superior performance for hybrid systems.
- Edge computing approaches, such as MSISCSOA in [9], reduce task delay by 21.38%, emphasizing latency-critical applications like IoT. In contrast, OMPA in [3] reduces missed deadlines, improving response times for mobile edge systems.

Similarities:

- A universal focus on reducing makespan and task delays.
- All algorithms optimize task dependencies, whether in DAG-based workflows, used in [1], or moldable workflows in [7].

Differences:

- Methods such as PPPS, presented in [6], while achieving a 93% reduction in execution time, also prioritize privacy, showing that execution time optimization is often coupled with other objectives.

Similarly, energy optimization emerges as a critical concern, particularly for edge and cloud systems:

- EPEE, presented in [5], stands out with significant reductions in energy consumption, leveraging DVFS to adapt workloads across geographically distributed data centers. This focus on energy efficiency is echoed by MSISCSOA in [9], which reduces energy use by 19.56%, balancing it with delay optimization.
- The marine-predator-based approach in OMPA, used in [3], introduces innovative methods for minimizing energy use while ensuring optimal workload distribution, aligning with the energy-saving principles of FARO in [2].

Similarities:

- Both edge (e.g., MSISCSOA in [9]) and cloud systems (e.g., EPEE in [5]) employ heuristic methods to balance energy and resource utilization.

Differences:

- Energy savings in edge systems, such as in [9], focus on reducing latency-related power usage, whereas cloud-centric approaches like in [5] emphasize operational cost savings tied to energy tariffs.

Cost reduction is another recurring objective, especially in hybrid and cloud environments:

- HSHC, presented in [8], achieves a 40% reduction in workflow costs, demonstrating the value of hybrid approaches that adapt resource allocations dynamically based on workload changes.
- GMPSO, used in [10], similarly balances cost and makespan, leveraging genetic enhancements to outperform traditional PSO and heuristic algorithms.

Similarities:

- Cost optimization is a shared focus in cloud systems (e.g., HSHC in [8] and GMPSO in [10]) and MEC environments (e.g., FARO in [2]), highlighting the relevance of balancing resource use with financial constraints.

Differences:

- Privacy-aware systems, such as PPPS in [6], address cost indirectly through WAN usage reduction rather than explicit cost optimization.

Privacy preservation is a unique dimension, primarily addressed in PPPS approach in [6], which minimizes WAN usage by 99% while ensuring compliance with data privacy regulations. This trade-off between performance and privacy is distinct from the goals of other studies, which do not explicitly address data protection.

- However, HSHC, presented in [8], and FARO in [2] share similarities with PPPS, used in [6], in their focus on data locality, reducing data transfer costs while optimizing task allocation.

Similarities:

- Data locality optimization is a shared focus, particularly in hybrid cloud environments (e.g., HSHC in [8]).

Differences:

- Privacy-specific objectives, such as those in [6], highlight a unique focus that is not present in energy- or cost-driven methods.

Efficient resource utilization is central to all studies, yet the methods vary significantly:

- MRSA in [7] optimizes multi-resource workflows by reallocating CPU, memory, and I/O dynamically, showing scalability in HPC environments.
- Similarly, EFaaS, used in [4], dynamically prioritizes tasks using a bidding mechanism, ensuring fairness in multi-user environments.

Similarities:

- Dynamic resource allocation is universally employed to address workload variability.

Differences:

- Resource-specific optimizations, such as multi-resource allocation in [7], differ from privacy-focused allocations in [6] or energy-aware distributions in [5].

5 Synthesis and Discussion

Several common strengths and innovative approaches emerge from the comparative analysis:

- Adaptability: Many algorithms incorporate mechanisms to adapt to dynamic workloads and resource availability, enhancing their effectiveness in heterogeneous and unpredictable environments.
- Multi-Objective Optimization: Studies like [3], [9], and [10] successfully balance multiple objectives, such as execution time, energy consumption, and cost, demonstrating the potential of advanced optimization techniques.
- Advanced Optimization Techniques: The use of machine learning, metaheuristic algorithms, and hybrid approaches represents a significant advancement over traditional scheduling methods, enabling more intelligent and efficient scheduling decisions.
- Scalability: Several algorithms demonstrate scalability to large workflows and diverse system sizes, highlighting their applicability to real-world scenarios with complex and extensive computational demands.

- Privacy and Security Considerations: Studies like [2] and [6] address the growing importance of data privacy and security, integrating these considerations into the scheduling process.

Despite common goals, the studies exhibit key differences and trade-offs:

- Focus Areas: Some studies prioritize execution time and latency ([1], [4], [9]), while others focus on energy efficiency ([3], [5], [10]) or privacy preservation ([6]).
- Optimization Techniques: The choice of optimization algorithm varies, with some studies employing reinforcement learning ([1]), others using metaheuristic algorithms ([3], [9], [10]), and some integrating hybrid approaches ([2], [8]).
- Trade-offs Between Objectives: Balancing multiple objectives often involves trade-offs. For example, focusing on energy efficiency might lead to increased execution times, while prioritizing execution speed could result in higher energy consumption or costs.

The studies demonstrate different strategies for balancing conflicting objectives:

- [3]: The OMPA algorithm combines workload prediction with opposition-based learning to minimize execution time and energy consumption while reducing resource wastage.
- [9]: The MSISCSOA algorithm effectively balances execution latency and energy consumption by employing a multi-strategy optimization approach, enhancing global optimization capabilities and preventing premature convergence.
- [10]: The GM-MOPSO algorithm integrates genetic operators into particle swarm optimization to optimize execution time and energy consumption simultaneously, achieving superior performance in HPC workflow scheduling.

The findings have practical implications across various computing environments:

- Edge Computing and MEC: Algorithms like [2], [3], and [9] are particularly relevant for edge computing and MEC environments, where resource constraints and latency requirements are critical.
- Cloud and Hybrid Environments: Methods in [5] and [8] offer valuable strategies for cloud service providers and organizations utilizing hybrid cloud infrastructures, enabling cost savings and improved performance through optimized resource allocation.
- HPC Systems: Studies like [1], [7], and [10] contribute to enhancing the efficiency of HPC systems, benefiting applications that require significant computational power and efficient workflow execution.
- Privacy-Sensitive Applications: [6] provides solutions for organizations dealing with sensitive data across multiple jurisdictions, ensuring compliance with data privacy regulations without compromising performance.

The studies also reveal limitations and areas for improvement:

- Computational Complexity: Some advanced algorithms may introduce significant computational overhead, potentially limiting their applicability in real-time or resource-constrained environments.

- Assumptions in Models: Certain models rely on accurate real-time data, such as electricity pricing ([5]) or workload predictions ([3]), which may not always be available or reliable.
- Scalability Challenges: While many algorithms demonstrate scalability, handling extremely large-scale systems or highly dynamic environments remains a challenge.
- Integration of Privacy and Security: Not all studies sufficiently address privacy and security considerations, highlighting the need for more comprehensive integration of these aspects into scheduling algorithms.

6 Future Research Directions

Building upon these findings, future research could focus on incorporating renewable energy sources into scheduling algorithms, further reducing environmental impact and operational costs. This integration would involve adapting scheduling strategies to consider the availability and variability of renewable energy, optimizing resource allocation accordingly.

Additionally, as data privacy regulations become increasingly stringent, developing advanced privacy-preserving techniques is essential. Future studies could explore integrating secure multi-party computation, differential privacy, or blockchain technologies into scheduling algorithms to enhance data protection.

Addressing scalability challenges requires developing algorithms that can efficiently handle larger workloads and more dynamic environments. Research could explore distributed optimization techniques, hierarchical scheduling models, and real-time adaptability to accommodate rapid changes in workload and resource availability.

Combining multiple optimization methods may yield more robust and efficient scheduling algorithms. Future research could investigate hybrid approaches that integrate machine learning, metaheuristic optimization, and heuristic methods to balance global exploration and local exploitation effectively.

Finally, emerging computing paradigms, such as quantum computing, edge-cloud continuum, and serverless architectures, present new challenges and opportunities for workflow scheduling. Future studies could explore how scheduling algorithms can be adapted or developed to leverage these technologies and address their unique constraints.

Conclusion

This literature review has provided a comprehensive comparative analysis of ten advanced workflow scheduling algorithms designed for HPC and cloud environments. The studies demonstrate significant advancements in addressing the complexities of workflow scheduling, offering innovative solutions that improve execution time, energy efficiency, cost optimization, privacy preservation, and resource utilization.

Common strengths among the studies include adaptability to dynamic workloads, scalability to large and heterogeneous systems, and the ability to balance multiple objectives through advanced optimization techniques. The use of machine learning, metaheuristic algorithms, and hybrid approaches reflects the evolving landscape of workflow scheduling research.

Key differences lie in the specific objectives prioritized, the optimization methods employed, and the application domains targeted. These differences highlight the importance of tailoring scheduling strategies to the unique requirements of different computing environments.

The limitations identified in the studies point to areas where further research is needed, such as enhancing scalability, integrating privacy and security considerations more comprehensively, and reducing computational complexity.

Advanced scheduling algorithms play a crucial role in optimizing workflows in HPC and cloud environments, contributing to improved system performance, resource management, and compliance with evolving regulations. As computing demands continue to grow and diversify, ongoing research and development in this field are essential to meet the challenges of modern computing infrastructures.

References

[1] G. P Koslovski, K. Pereira, and P. R. Albuquerque, "DAG-based workflows scheduling using Actor–Critic Deep Reinforcement Learning," *Future Generation Computer Systems*, vol. 150, pp. 354-363, Jan. 2024, doi: 10.1016/j.future.2023.09.018.

[2] D. K. Sajnani, X. Li, and A. R. Mahesar, "Secure workflow scheduling algorithm utilizing hybrid optimization in mobile edge computing environments," *Computer Communications*, vols. 226–227, Art. no. 107929, Aug. 2024, doi: 10.1016/j.comcom.2024.107929.

[3] F. Kuang, Z. Xu, and M. Masdari, "Multi-workflow scheduling and resource provisioning in Mobile Edge Computing using opposition-based Marine-Predator Algorithm," *Pervasive and Mobile Computing*, vol. 87, Art. no. 101715, Dec. 2022, doi: 10.1016/j.pmcj.2022.101715.

[4] S. H. Mahdizadeh, S. Abrishami, "An assignment mechanism for workflow scheduling in Function as a Service edge environment," *Future Generation Computer Systems*, vol. 157, pp. 543-557, Aug. 2024, doi: 10.1016/j.future.2024.04.003.

[5] M. Hussain, L.-F. Wei, A. Rehman, A. Hussain, M. Ali, and M. H. Javed, "An electricity price and energy-efficient workflow scheduling in geographically distributed cloud data centers," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 8, Oct. 2024, doi: 10.1016/j.jksuci.2024.102170.

[6] Y. Xiao, A. C. Zhou, X. Yang, and B. He, "Privacy-preserving workflow scheduling in geo-distributed data centers," *Future Generation Computer Systems*, vol. 130, pp. 46-58, May 2022, doi: 10.1016/j.future.2021.12.004.

[7] L. Perotin, S. Kandaswamy, H. Sun, and P. Raghavan, "Multi-resource scheduling of moldable workflows," *Journal of Parallel and Distributed Computing*, vol. 184, Art. no. 104792, Feb. 2024, doi: 10.1016/j.jpdc.2023.104792

[8] A. Pasdar, Y. C. Lee, and K. Almi'ani, "Hybrid scheduling for scientific workflows on hybrid clouds," *Computer Networks*, vol. 181, Art. no. 107438, Nov. 2020, doi: 10.1016/j.comnet.2020.107438

[9] P. Jayalakshmi, S. S. Subashka Ramesh, "Multi-strategy improved sand cat optimization algorithm-based workflow scheduling mechanism for heterogeneous edge computing environment," *Sustainable Computing: Informatics and Systems*, vol. 43, Art. no. 101014, Sep. 2024, doi: 10.1016/j.suscom.2024.101014.

[10] H. Hafsi, H. Gharsellaoui, and S. Bouamama, "Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling," *Applied Soft Computing*, vol. 122, Art. no. 108791, Jun. 2022, doi: 10.1016/j.asoc.2022.108791

Application A. Comparison of Methods

Table 1. Comparison of Scheduling Methods

Research paper	Computing environment	Scheduling method	Optimization algorithm	Objective Function	Initial Data	Key method features
DAG-based workflows scheduling using Actor–Critic Deep Reinforcement Learning [1]	HPC Data Centers	Actor-Critic RL	Deep Reinforcement Learning (DRL)	Minimize task delay and maximize throughput	DAG-based workflows	Dynamic policy selection from existing algorithms
Secure workflow scheduling algorithm utilizing hybrid optimization in mobile edge computing environments [2]	Mobile Edge Computing (MEC)	Feedback Artificial Tree (FAT)	Remora Optimization Algorithm (ROA)	Minimize CPU, memory usage, and encryption cost	MEC resources and tasks	Hybrid approach for enhanced security and efficiency
Multi-workflow scheduling and resource provisioning in Mobile Edge Computing using opposition-based Marine-Predator Algorithm [3]	Mobile Edge Computing (MEC)	Marine Predator Algorithm (MPA)	Opposition-based Marine Predator Algorithm (OMPA)	Reduce missed deadlines, minimize VMs	Historical IoT data in MEC	Uses opposition-based learning to avoid local minima
An assignment mechanism for workflow scheduling in Function as a Service edge environment [4]	Edge Function as a Service (FaaS)	Highest Bid First (HBFM), Warm Function First (WFFM)	Bidding + Priority Mechanisms	Minimize makespan	EFaaS resources	Bidding-based and priority assignment mechanisms
An electricity price and energy-efficient workflow scheduling in geographically distributed cloud data centers [5]	Geo-distributed cloud data centers	Task ranking and data center selection	Dynamic Voltage and Frequency Scaling (DVFS)	Minimize electricity costs	Geo-distributed cloud data	Uses DVFS and variable energy tariffs
Privacy-preserving workflow scheduling in geo-distributed data centers [6]	Geo-distributed DCs	Privacy-Preserving Graph Partitioning	Privacy-Aware Refinement	WAN usage and privacy adherence	Geo-distributed DCs with privacy levels	Two-stage privacy-preserving workflow scheduling
Multi-resource scheduling of moldable workflows [7]	HPC systems	Multi-resource optimization	MRSA (Resource-aware Scheduling)	Reduce makespan while preserving data privacy	Moldable workflows	Enables pre-execution resource adjustments
Hybrid scheduling for scientific workflows on hybrid clouds [8]	Hybrid clouds	Two-phase Scheduling (Static/ Dynamic)	HSHC (Genetic + Dynamic Adjustment)	Reduce cost, improve time	Scientific workflows	Handles data locality dynamically
Multi-strategy improved sand cat optimization algorithm-based workflow scheduling mechanism for heterogeneous edge computing environment [9]	Hybrid Cloud-Edge Computing	Sand Cat Optimization (SCOA)	MSISCSOA (Heuristic Swarm Optimization)	Minimize delay and energy consumption	Edge computing resources	Multi-strategy approach with dynamic search
Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling [10]	Hybrid cloud + HPC	Task Mapping via Matrix Encoding	GMPSO (Genetic + PSO)	Optimize cost and makespan	HPC workflows	Introduces genetic operations into PSO

Application B. Comparison of Results

Table 2. Comparison of Performance Results

Research paper	**Perfomance metrics**	**Key result**	**Experimental Data**
DAG-based workflows scheduling using Actor–Critic Deep Reinforcement Learning [1]	Makespan, throughput, task delay	35% improvement over FCFS and SPT in DAG processing speed	Simulations with synthetic and real DAGs
Secure workflow scheduling algorithm utilizing hybrid optimization in mobile edge computing environments [2]	CPU utilization, memory usage, encryption cost, execution time	CPU usage reduced to 0.012, memory to 0.010, encryption cost minimized	Workflows with varying task sizes (100, 200, 300 tasks)
Multi-workflow scheduling and resource provisioning in Mobile Edge Computing using opposition-based Marine-Predator Algorithm [3]	Makespan, deadline miss rates, resource utilization	Significant reductions in makespan and deadline misses, improved utilization	iFogSim with NASA and Saskatchewan datasets
An assignment mechanism for workflow scheduling in Function as a Service edge environment [4]	Normalized Completion Time (NCT), resource utilization	Significant improvements in execution times and reduced cold start delays	Simulated EFaaS workflows with priority mechanisms
An electricity price and energy-efficient workflow scheduling in geographically distributed cloud data centers [5]	Energy consumption, operational costs, execution time	Energy consumption reduced by up to 25%, costs decreased by 20-30%	Simulations in CloudSim with synthetic and real workflows
Privacy-preserving workflow scheduling in geo-distributed data centers [6]	Execution time, WAN usage, privacy compliance	Execution time reduced by up to 93%, WAN usage by up to 99%	Simulations with real-world workflows (MONTAGE, PSLOAD, PSMERGE) and Azure traces
Multi-resource scheduling of moldable workflows [7]	Workflow completion time, resource utilization	Up to 30% reduction in execution time, utilization above 85%	Simulations with synthetic workloads and HPC benchmarks
Hybrid scheduling for scientific workflows on hybrid clouds [8]	Execution cost, execution time, deadline compliance	Cost reduced by 40%, execution time improved by 25%	Tests with real-world and synthetic workflows in hybrid clouds
Multi-strategy improved sand cat optimization algorithm-based workflow scheduling mechanism for heterogeneous edge computing environment [9]	Execution latency, energy consumption	Reduced latency by 21.38%, energy consumption by 19.56%	Simulations using iFogSim with scientific workflows
Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling [10]	Execution time, energy consumption, cost efficiency	Energy consumption reduced by up to 30%, execution time improved by 25%	Benchmark workflows (Cybershake, Epigenomics, Montage)