Review Article

Waleed Kareem Awad*, Khairul Akram Zainol Ariffin, Mohd Zakree Ahmad Nazri, and Esam Taha Yassen

Resource allocation strategies and task scheduling algorithms for cloud computing: A systematic literature review

https://doi.org/10.1515/jisys-2024-0441
received November 04, 2024; accepted February 12, 2025

Abstract: The concept of cloud computing has completely changed how computational resources are delivered and used. By enabling on-demand access to collective computing resources through the internet. While this technological shift offers unparalleled flexibility, it also brings considerable challenges, especially in scheduling and resource allocation, particularly when optimizing multiple objectives in a dynamic environment. Efficient allocation and scheduling of resources are critical in cloud computing, as they directly impact system performance, resource utilization, and cost efficiency in dynamic and heterogeneous conditions. Existing approaches often face difficulties in balancing conflicting objectives, such as reducing task completion time while staying within budget constraints or minimizing energy consumption while maximizing resource utilization. As a result, many solutions fall short of optimal performance, leading to increased costs and degraded performance. This systematic literature review (SLR) focuses on research conducted between 2019 and 2023 on scheduling and resource allocation in cloud environment. Following preferred reporting items for systematic reviews and meta-analyses guidelines, the review ensures a transparent and replicable process by employing systematic inclusion criteria and minimizing bias. The review explores key concepts in resource management and classifies existing strategies into mathematical, heuristic, and hyper-heuristic approaches. It evaluates popular algorithms designed to optimize key metrics such as energy consumption, resource utilization, cost reduction, makespan minimization, and performance satisfaction. Through a comparative analysis, the SLR discusses the strengths and limitations of various resource management schemes and identifies emerging trends. It underscores a steady growth in research within this field, emphasizing the importance of developing efficient allocation strategies to address the complexities of modern cloud systems. The findings provide a comprehensive overview of current methodologies and pave the way for future research aimed at tackling unresolved challenges in cloud computing resource management. This work serves as a valuable resource for practitioners and academics seeking to optimize scheduling and allocation in dynamic cloud environments, contributing to advancements in resource management strategies of cloud computing.

Keywords: resource allocation, task scheduling, heuristic approach, hyper heuristic approach and mathematical approach

*** Corresponding author: Waleed Kareem Awad,** Data Mining and Optimization Research Group (DMO), Centre for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600, Bandar Baru Bangi, Malaysia; College of Computer Science and Information Technology, University of Anbar, Al Anbar, 31001, Iraq, e-mail: waleed.kareem@uoanbar.edu.iq, p131629@siswa.ukm.edu.my
Khairul Akram Zainol Ariffin: Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600, Bandar Baru Bangi, Malaysia, e-mail: k.akram@ukm.edu.my
Mohd Zakree Ahmad Nazri: Data Mining and Optimization Research Group (DMO), Centre for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600, Bandar Baru Bangi, Malaysia, e-mail: zakree@ukm.edu.my
Esam Taha Yassen: College of Computer Science and Information Technology, University of Anbar, Al Anbar, 31001, Iraq, e-mail: co.esamtaha@uoanbar.edu.iq

1 Introduction

In recent years, cloud computing has emerged as a crucial instrument of the information technology industry. It offers scalability, flexibility, and cost-effectiveness through on-demand computing services that let customers access resources online [1]. However, it faces significant challenges in resource allocation and scheduling that extensively affect its performance and efficiency [2]. One of the primary issues is the dynamic nature of user demands, which can fluctuate significantly over time. This variability necessitates advanced algorithms for effective resource management, ensuring that computational resources are allocated optimally to meet varying workloads. Failure to accurately predict and respond to these fluctuations can lead to underutilization or overloading of resources, resulting in performance degradation and increased operational costs. Effective resource allocation must account for these limitations while making certain that tasks are completed within the required timeframes and efficiently [3].

To address these challenges, the researchers have explored various approaches, ranging from traditional static heuristics to more advanced artificial intelligence (AI)-driven methods. Traditional methods often rely on static heuristics, where decisions are made based on predefined rules or empirical guidelines [4]. AI-driven methods, such as metaheuristic (MH) algorithms, have gained popularity due to their adapting ability to dynamic conditions and optimize multiple objectives simultaneously [5]. Furthermore, the integration of machine learning techniques into scheduling and resource management has opened new avenues for enhancing efficiency and performance in cloud computing [6]. Models of machine learning can analyze historical data to predict workloads and optimize resource allocation more accurately than traditional heuristics. This predictive capability allows for proactive adjustments in resource distribution, minimizing latency, and maximizing throughput [7].

Recent studies show that the reinforcement learning (RIL) is effective for task scheduling, with agents learning optimal policies through trial and error in a simulated cloud environment [8]. These adaptive strategies not only improve immediate task performance; but also contribute to long-term resource optimization by learning from previous decisions and their outcomes. Moreover, the emergence of hybrid models, which combine the strengths of both traditional and AI-driven methodologies, has shown promising results [9]. By leveraging the simplicity of static heuristics alongside the adaptability of machine learning algorithms, these hybrid approaches can provide more robust solutions under varying conditions [10]. The static heuristics shift to advanced AI-driven methods which represents significant progress in managing the complexities of cloud computing. Ongoing research and development are essential for optimizing task scheduling and resource management, ultimately enhancing service quality and user satisfaction in cloud infrastructures [11].

Despite these progressions, resource allocation and task scheduling methods still have significant gaps. First, static heuristics, while straightforward, are often inadequate for handling the dynamic and unpredictable natures of cloud workloads. They lack the flexibility to adapt to the changes in real-time resource availability or user demands, leading to suboptimal performance in dynamic environments [5]. Second, their practical implementation faces challenges related to scalability and computational complexity, while nature-inspired techniques such as genetic algorithms (GAs) and particle swarm optimization (PSO) show promise. These methods often require significant computational resources and time, making them less feasible for large-scale cloud environments [10]. Third, many current studies had focus on single performance metrics, such as energy efficiency or execution speed, without providing a comprehensive evaluation of multiple dimensions, such as fairness, user-centric priorities, and adaptability [12]. This limited emphasis making these techniques less applicable in real-world situations when juggling several goals at once is so necessary. Fourth, there are insufficient standardized experimental platforms and simulation tools for evaluating task scheduling and resource allocation strategies, making it difficult to compare results across studies [13]. Finally, the rapid evolution of cloud computing technologies and workloads necessitates continuous updates to existing methodologies, yet many studies fail to address emerging trends or provide forward-looking insights [14].

In this systematic literature review (SLR), the latest articles related to resource allocation and task scheduling in clouds have been reviewed. We have investigated and categorized a large number of related articles based on their objectives, characteristics, and simulation tools. By identifying and critically analyzing

the shortcomings of current approaches, this SLR provides a roadmap for future research and development. Additionally, it highlights new trends and possible future paths in cloud computing resource management by synthesizing lessons from a variety of studies. By adhering to the preferred reporting items for systematic reviews and meta-analyses (PRISMA) guidelines and employing a rigorous selection process, this SLR aims to provide a high-quality, unbiased synthesis of the research of current state in this field. To guide this SLR and address key aspects of scheduling and allocation issues in cloud computing, the following research inquiries were formulated:

RQ1. What are the primary challenges in cloud computing environments of resource allocation and task scheduling?

RQ2. How do various optimization methods perform in dynamic cloud environments?

RQ3. What are the shortcomings and difficulties of the current task scheduling algorithms and resource allocation strategies?

RQ4. How do mathematical, heuristic, and hyper-heuristic methods compare in terms of efficiency and scalability for task scheduling in cloud environments?

RQ5. Which experimental platforms or simulation tools are utilized to assess task scheduling and resource scheduling strategies in cloud computing?

RQ6. What are the emerging trends and future directions in this area of research?

By addressing these questions, this SLR seeks to highlight key features and provide insights into future resource allocation trends and cloud computing task scheduling trends. It aims to offer a holistic overview of the most recent algorithms and approaches, which will aid future academics in advancing research in this critical area of cloud computing.

This SLR seeks to tackle these significant gaps by providing a comprehensive and structured analysis of resource allocation and task scheduling strategies from 2019 to 2023. Specifically, this review contributes to the field in the following ways:

- Conducting a structured review of resource allocation and task scheduling strategies from 2019 to 2023.
- Introducing a new taxonomy that categorizes approaches into mathematical, heuristic, and hyper-heuristic methods.
- Conducting an in-depth comparative analysis that evaluates methods across multiple performance dimensions, including efficiency, scalability, and adaptability.
- Identifying and critically analyzing the shortcomings of current approaches, thus providing a roadmap for future research and development.
- Synthesizing insights from a wide range of studies to highlight emerging trends and potential future directions in cloud computing resource management.

The study is structured as follows: Section 2 discusses related work. Section 3 outlines the research methodology. Section 4 discusses resource allocation and task scheduling in cloud computing. Section 5 reviews the current state of resource management approaches. Section 6 provides a discussion. Section 7 presents the study characteristics. Section 8 offers challenges and research gaps, and the final section concludes the study.

2 Related work

The challenge of resource allocation and task scheduling in cloud computing is a well-known NP-hard challenge that has garnered significant attention from researchers. Despite extensive studies, it remains a compelling area of research due to the dynamic and complex nature of cloud environments [5]. Developing efficient scheduling solutions using existing optimization algorithms continues to be a challenge, given the rapidly evolving demands and constraints of cloud computing [15]. This section evaluates the outcomes of previous reviews on optimization algorithms, emphasizing their strengths and weaknesses to provide a comprehensive understanding of their effectiveness in addressing this critical issue.

Jia et al. [16] conducted a systematic review of multi-tenancy scheduling approaches in cloud platforms, examining scheduling policies, cloud provisioning, and deployment. Their classification system encompasses static, complex, offline, online, preemptive, and non-preemptive scheduling algorithms. However, the study lacks specific details on the limitations of the reviewed scheduling approaches on multi-tenancy cloud platforms. Negi et al. [17] conducted a comprehensive analysis of cloud load-balancing techniques using computational paradigms. The study categorizes methods based on soft computing approaches including fuzzy systems, machine learning, neural networks, and bio-inspired computing, evaluating their effectiveness at both virtual machine (VM) and physical machine levels. However, the study lacks a detailed analysis of the performance metrics for evaluating the effectiveness of soft computing techniques in achieving dynamic load balancing.

A significant systematic review of MH algorithms for cloud task scheduling was illustrated by Houssein et al. [18]. The study addresses resource utilization challenges and offers valuable categorization based on scheduling problem nature, objectives, task-resource mapping schemes, and constraints, while highlighting the importance of efficient task distribution across limited resources. Another significant survey on task scheduling techniques in cloud computing was presented by Panwar et al. [11]. The stud explores energy-saving strategies for cloud data centers, addressing the pressing challenges of high electricity consumption and environmental impact. The researchers systematically evaluated various approaches, including machine learning, heuristics, MHs, and statistical methods to enhance resource management and energy utilization. Their findings demonstrated notable energy saving compared to conventional techniques. However, while the study contributes value to energy efficiency research, it does not comprehensively address other crucial aspects of cloud data center operations, such as security measures, system scalability, and service reliability.

Zhou et al. [19] provide a contrasting analysis of MH load-balancing algorithms in cloud computing, evaluating the factors of performance including makespan time, degree of imbalance, data center processing time, flow time, response time, and resource utilization. The study examines challenges in integrating and improving MH methods for load balancing, such as reconfiguring transformation operators, extracting features from input workloads, and implementing hybrid models. However, the study lacks a detailed discussion of specific challenges and limitations for each analyzed algorithm. It also omits critical analysis of potential drawbacks in applying MH methods to load balancing, particularly regarding computational complexity and convergence issues.

Zhou et al. [19] highlighted the significance of task scheduling algorithms in cloud computing, with particular focus on the round robin (RR) algorithm and its enhancements. The RR algorithm is commonly used due to its simplicity and time-sharing capabilities. While it effectively addresses efficient resource utilization and task scheduling challenges, the RR algorithm may still face performance optimization issues, especially in dynamic cloud computing environments. Also, an important comprehensive review of nature-inspired scheduling approaches, evaluating their effectiveness based on qualitative Quality of Service (QoS) parameters and simulation tools in cloud environments was presented by Arunarani et al. [20]. Algorithms, such as henry gas solubility optimization and cat swarm optimization, have demonstrated effectiveness in balancing exploitation and exploration while preventing local optima. These approaches optimize various criteria such as makespan and resource utilization, though certain limitations exist in specific cloud computing scenarios.

Subsequent research [19,21–24] covers various aspects of cloud computing, including priority-based scheduling, MH load balancing, job management techniques, and resource management. Significant contributions include comparative analyses of performance metrics, taxonomies of resource allocation approaches, and evaluations of hybrid algorithms. Collectively, these studies enhance our understanding of cloud computing optimization while highlighting areas needing further research, such as specific technical limitations, computational complexity considerations, and integration challenges.

This review is distinguished by its methodological rigor, time-bound focus, comprehensive classification, and attention to multi-objective optimization in current cloud computing challenges. The focused timeframe (2019–2023) provides current insights into contemporary cloud computing challenges, while the unique classification of methods into mathematical, heuristic, and hyper-heuristic methods provides a more complete theoretical framework compared to previous reviews that often focus on single aspects such as nature-inspired approaches, or energy efficiency. In contrast to previous studies that tackle single objectives, our focus on balancing conflicting objectives, such as energy efficiency and load balancing, better reflects the complexity

of real-world scenarios, which earlier studies often simplify by addressing single objectives. Furthermore, this review also integrates modern challenges, emphasizing dynamic and heterogeneous conditions.

3 Research methodology

This SLR employs an exploratory and descriptive procedure to meet the research's objectives and address the questions raised in the introduction section. This approach facilitates an understanding of the state-of-the-art in the field, identifies previous studies limitations, and emphasizes endorsements for future research. This SLR was systematically carried out in conformity with the PRISMA 2020 statement and established methodological guidelines [25,26]. Its reporting included methods and materials, inclusion and exclusion criteria, search strategies, information sources, risk of bias assessment, data management, and the PRISMA flow chart that represents the overall methodological design.

3.1 Inclusion and exclusion criteria

In this step, systematic and comprehensive reviews of relevant literature were conducted to determine inclusion and exclusion criteria for this study. The inclusion criteria refer to all articles selected for examination during the procedure to perform a systematic review. This selection is based on a strategic relationship with keywords such as "cloud computing," "resource allocation," and "task scheduling." The inclusion criteria encompass peer-reviewed journal articles produced from 2019 to 2023 that focus on resource allocation and task scheduling in cloud computing, utilizing mathematical, heuristic, or hyper-heuristic methods.

The purpose of the exclusion criteria was to exclude studies misaligned with the research objectives during the SLR process, as guided by the PRISMA 2020 statement. The first step in the research selection process was to search for literature sources. After removing duplicates, a three-iteration screening and filtering process was conducted:

- First iteration: Articles were excluded according to their titles and abstracts if they did not address the predefined research topics.
- Second iteration: Articles were excluded if they were inaccessible due to restricted access, closed-access journals, or other barriers.
- Third iteration: Articles were excluded if they did not focus on resource allocation and task scheduling in a cloud computing environment or did not adopt one of the specified approaches: mathematical, heuristic, or hyper-heuristic.

3.2 Data selection source

In January 2024, a search for articles was conducted. Following the PRISMA statement for literature reviews, it is crucial to specify the information sources and the inclusion and exclusion criteria used for the analysis. This SLR examined all publications available in relevant databases from 2019 to 2023. A systematic and comprehensive search was performed in five academic databases: ScienceDirect, Web of Science, Springer, IEEE Xplore, and Scopus, focusing on research publications related to resource allocation and scheduling in cloud computing. Publications from 2019 to 2023 were reviewed and analyzed; 2024 was removed as incomplete. Considering the timing of this SLR, the term "PUBYEAR > 2018 AND PUBYEAR < 2024" was used to provide access to the relevant publications. We also utilized search options to exclude book chapters, blogs, theses, and other non-article formats, prioritizing peer-reviewed journal articles in English to ensure the quality and relevance of our findings.

3.3 Data search strategy

We sought resource allocating and scheduling methods in cloud environments. We created a consistent search strategy using predefined search terms based on the eligibility criteria to ensure that the studies from the five selected databases were relevant to the research objectives and compatible with each database's search interface. Searches were conducted across the five digital libraries using a search string "cloud computing" OR "cloud environment" with the subsequent keywords "Resource allocation" and "Task scheduling." Also, we used these keywords: "Mathematical approach," OR "Heuristic approach" OR "Hyper approach" to limit our scope of searches as shown in Table 1.

Table 1: Query in digital libraries used in this study

Digital library	Query/search terms	No. of articles
ScienceDirect	("cloud computing" OR "cloud environment") AND (("resource allocation" AND "task scheduling") OR ("resource allocation" OR "task scheduling")) AND ("Mathematical approach" OR "Heuristic approach" OR "Hyper Heuristic approach")	201
Web of Science	TS = ("cloud computing" OR "cloud environment") AND TS = (("resource allocation" AND "task scheduling") OR ("resource allocation" OR "task scheduling")) AND TS = ("Mathematical approach" OR "Heuristic approach" OR "Hyper Heuristic approach")	21
Springer	("cloud computing" OR "cloud environment") AND (("resource allocation" AND "task scheduling") OR ("resource allocation" OR "task scheduling")) AND ("Mathematical approach" OR "Heuristic approach" OR "Hyper Heuristic approach")	127
IEEE	(cloud computing) AND ((resource allocation AND task scheduling) OR (resource allocation OR "task scheduling)) AND (Mathematical approach OR Heuristic approach OR Hyper Heuristic approach)	205
Scopus	("cloud computing" OR "cloud environment") AND (("resource allocation" AND "task scheduling") OR ("resource allocation" OR "task scheduling")) AND ("Mathematical approach" OR "Heuristic approach" OR "Hyper Heuristic approach")	367
Total number of articles collected in the initial search		921

3.4 Risk of bias assessment

A well-defined and consistent procedure was implemented to evaluate the bias risk in the studies included in this review. Each author contributed to the data collecting and bias risk assessment processes to make sure of the accuracy and results integrity. A Microsoft Excel based automated tool was utilized to facilitate an impartial and consistent evaluation. The authors independently evaluated each article and then worked together to resolve any difficulties or conflicts until they were all in accord.

To ensure the reliability of the results, transparent and consistent criteria were applied in assessing the risk of bias. However, it is important to note that the study was limited to articles retrieved from five databases: ScienceDirect, Web of Science, Springer, IEEE Xplore, and Scopus. This limitation may have introduced bias by potentially excluding relevant research available in other databases or sources related to task scheduling and resource allocation. Despite this constraint, efforts were made to minimize bias by adhering to strict inclusion, exclusion principles and undertaking a comprehensive search across the selected databases.

3.5 Data management and quality control

Using the search algorithms available in each database, 921 empirical studies on scheduling and resource allocation in cloud computing were initially retrieved. Among these, 201 studies were from ScienceDirect, 21 from Web of Science, 127 from Springer, 205 from IEEE Xplore, and 367 from Scopus databases. Owing to the various typological formats of the databases utilized, a data homogenization procedure was performed on the

studies in Microsoft Excel to unify the format. To address the study questions that were given, the same technology was utilized to analyze the data and apply the exclusion criteria.

3.6 Data selection process

In this section, based on the search process we conducted, we obtained 921 articles from scientific digital libraries as a primary search. Eighty-seven duplicate articles were eliminated, reducing the number to 834. Afterward, 368 articles were excluded due to title and abstract criteria. Furthermore, 282 documents were inaccessible due to publisher restrictions, required institutional subscriptions, or limited online availability. Finally, after reviewing the remaining 184 articles, 84 were excluded based on full-text criteria, leaving 100 studies included in this review. Figure 1 shows the PRISMA flow diagram. Figure 2 shows the number of publications per year in the cloud environment. Figure 3 illustrates the contribution of each methodology/application to the publications over time.

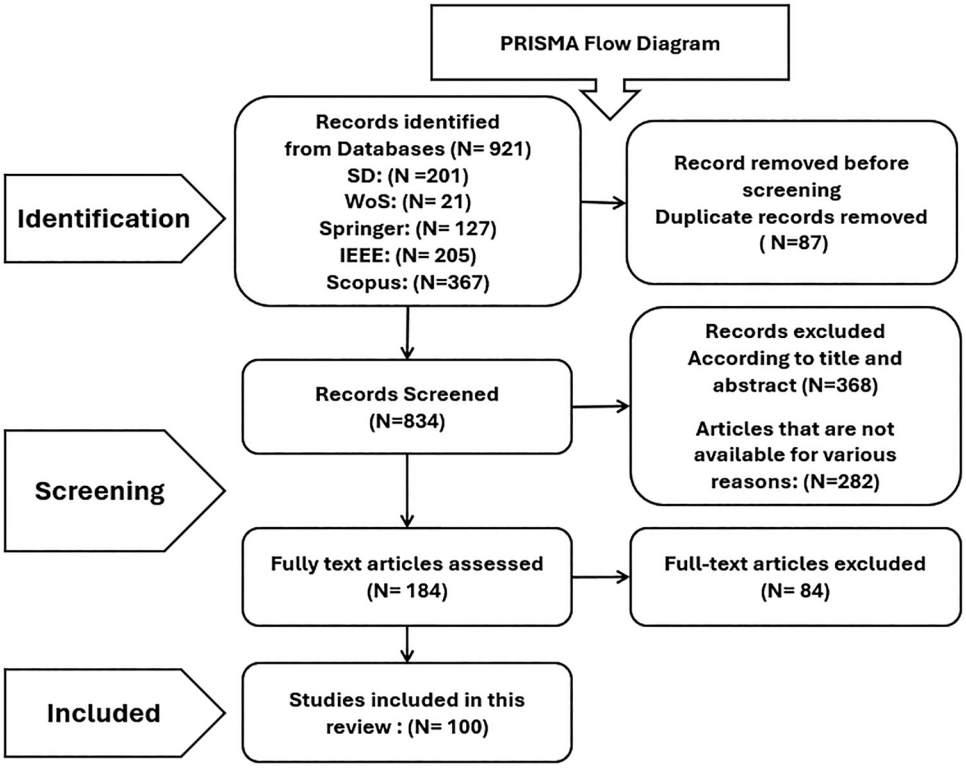

Figure 1: The PRISMA flow diagram (created by the authors).

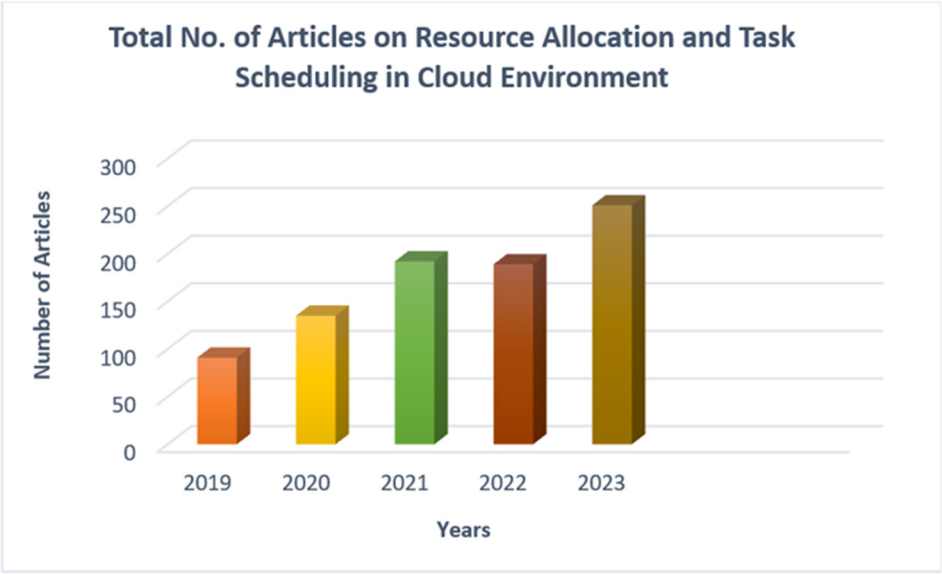

Figure 2: Rate of publications on resource allocation and task scheduling from 2019 to 2023 according to our systematic review (created by the authors).

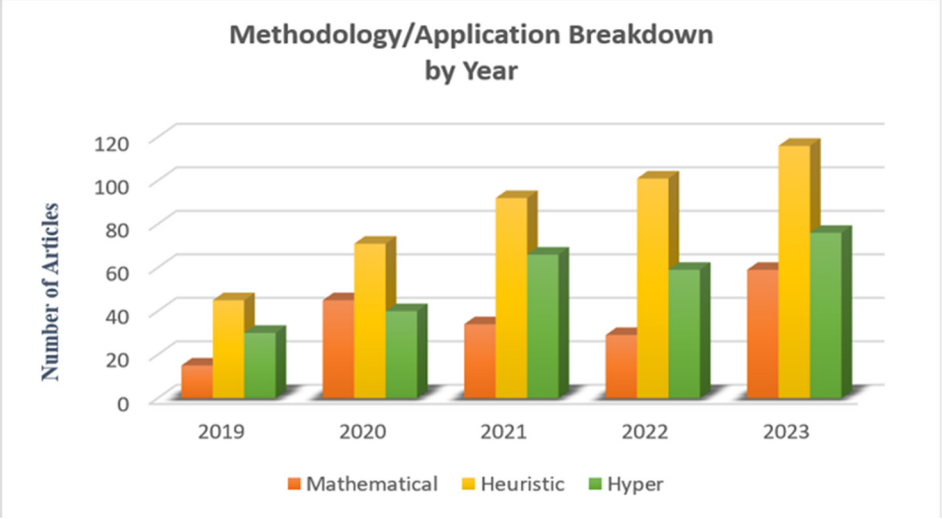

Figure 3: The contribution of each methodology/application to the publications over time (created by the authors).

3.7 Impact of study quality and bias on review results

The quality of the studies included in this SLR significantly impacts the reliability and validity of the review results. The reviewed studies exhibit variability in design and methodological rigor, with many relying heavily on simulation-based approaches such as CloudSim, which introduces simulation bias and limits the generalizability of the findings to real-world cloud computing environments. Additionally, the frequent use of small synthetic datasets, which may not reflect the diversity and variability of actual cloud workloads, results in dataset bias and potentially skews algorithm performance evaluations.

Metric selection bias is shown by the emphasis on traditional metrics such as makespan at the expense of other essential aspects such as user satisfaction, security, dependability, or energy efficiency, which restricts a thorough knowledge of algorithm performance. It is difficult to compare and synthesize findings across

research due to the absence of common evaluation standards and benchmarks, further complicating the process of coming to consistent conclusions or suggestions. Utilizing supportive tools such as Microsoft Excel and Cloude makes this analysis easier, but it is vital to note that the authors' verification is necessary to guarantee the integrity and quality of the data. Overall, the review underscores the need for more rigorous study designs, synthetic datasets, and diverse evaluation metrics to improve the caliber and relevance of studies on scheduling and allocation, ensuring that findings are reliable and relevant to real-world scenarios.

4 Allocation of resource and scheduling of task in cloud computing

In light of the rapid growth and evolution of cloud computing, the efficient allocation of resource and the judicious scheduling of tasks have become critical to improve performance and reduce costs. As organizations increasingly rely on cloud infrastructures to meet their computing requirements, understanding the intricacies of resource management has become critical [27]. These processes are fundamental and vital for optimizing the utilization of cloud resources while simultaneously meeting user requirements and provider objectives [28].

4.1 Definitions and importance

Resource allocation denotes the distribution of computing resources (such as CPU, memory, and storage) among various tasks or applications in a cloud-based system [29]. Task scheduling involves the process of assigning these tasks to the allocated resources in an optimal manner. Effective resource allocation and task scheduling are essential for various reasons [30]:

1. Performance optimization: Proper resource allocation and task scheduling can significantly improve system performance and reduce task completion times.
2. Cost efficiency: Efficient resource utilization helps minimize operational costs for cloud providers and users.
3. QoS: It ensures adherence to service level agreements (SLAs), guaranteeing consistency, timeliness, and reliability.
4. Energy efficiency: Optimized resource usage can minimize energy consumption in the centers of cloud data.

4.2 Challenges in cloud resource management

The complexity of resource allocation and scheduling task in cloud environments stems from several factors [31]:

1. Heterogeneity: Cloud resources and user tasks are diverse, making finding optimal matches challenging.
2. Dynamism: Cloud environments' workload and resource availability constantly change.
3. Conflicting objectives: Cloud providers aim to maximize profits and resource utilization, while users seek to minimize costs and execution times.
4. NP-hard nature: The resource allocation and scheduling problem in cloud computing is NP-hard, making it computationally intensive to find optimal solutions.

4.3 Key entities and their objectives

Cloud computing involves two primary entities [32]: The essential interaction of key entities in cloud computing can be easily understood using Figure 4.

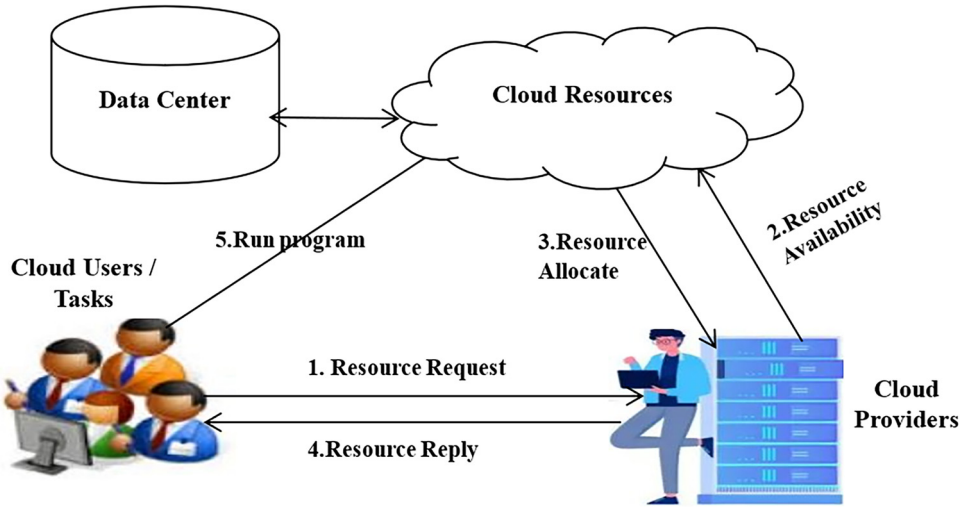

Figure 4: The basic interaction flow between cloud entities (created by the authors).

1. Cloud providers:
- Establish and maintain cloud data centers
- Provide computing resources on a rental basis
- Aim to maximize profit through optimal resource utilization

2. Cloud users (consumers):
- Utilize cloud resources for their applications
- Seek to run applications within set time and budget constraints

The main interaction is shown in the following steps [33,34]: A cloud user submits a request (task) for any specific resource to the cloud provider. Upon receiving the request, the cloud provider verifies resource availability. If resources are available, they are allocated to the requesting user, who then utilizes the resources to execute the desired activity or application. If the users no longer require the resources, they release them [35]. Then, the provider schedules and allocates resources to the other requesting clients. The allocation of resources and scheduling tasks in cloud computing is a widely studied optimization problem, making it a highly appealing research area. However, achieving efficient scheduling using state-of-the-art optimization algorithms remains challenging due to the inherent complexity and dynamic nature of cloud computing [5].

Allocation strategies and scheduling methods in cloud computing environments typically optimize various parameters, which can be broadly categorized into consumer-centric and provider-centric goals [36]. Consumer-centric objectives focus on minimizing execution time (makespan), reducing costs, and meeting deadline constraints, all aimed at enhancing user satisfaction and operational efficiency [37]. On the other hand, provider centric goals emphasize maximizing resource utilization, balancing workload across available resources, and minimizing energy consumption [38]. Achieving these goals is essential for ensuing operational efficiency and profitability. The primary challenge lies in balancing these often conflicting objectives to develop effective and holistic solutions [39].

The reviewed literature encompasses several critical subdomains and application scenarios within resource allocation and task scheduling in cloud computing. Edge computing, a prominent area, focuses on latency-sensitive applications such as Internet of Things (IoT) networks and real-time analytics, emphasizing reduced latency and efficient scheduling in resource-constrained environments [40]. Real-time applications, including video streaming and live data processing, require strategies that optimize response times and ensure consistent service under dynamic workloads [41]. Big data processing, decisive for data mining and large-scale analytics, prioritizes fault tolerance and throughput across distributed systems [42]. Energy-efficient cloud systems target sustainability by minimizing energy consumption while balancing performance metrics

such as makespan and resource utilization, especially in data centers and containerized environments [43]. Finally, hybrid and multi-cloud environments pose challenges in inter-cloud resource scheduling and load balancing, with strategies focusing on cost optimization, fault tolerance, and workload distribution across heterogeneous infrastructures [30]. These diverse subdomains highlight the breadth and complexity of resource management challenges in cloud computing, providing valuable insights for targeted improvements in future research.

4.4 Impact of IoT, AI, and quantum computing on resource allocation and scheduling

The integration of cloud computing with emerging technologies such as IoT, AI, and quantum computing is significantly influencing resource allocation and scheduling processes. IoT devices generate massive amounts of data that need to be processed and stored efficiently. Effective resource allocation and scheduling are crucial for handling these data deluge [5,40]. Techniques such as fog computing and edge computing are employed to process data closer to the source, emphasizing reduced latency and efficient scheduling in resource-constrained environments [44]. Research has shown that task scheduling mechanisms significantly influence resource allocation in IoT environments.

AI plays a significant role in optimizing resource allocation and scheduling. AI algorithms can predict workload patterns, automate resource provisioning, and optimize task scheduling to improve efficiency and reduce costs. For instance, AI can dynamically allocate resources based on real-time demand, ensuring optimal utilization. Furthermore, AI enhances automation in task scheduling, leading to more streamlined and cost-effective resource management [14].

Quantum computing introduces new paradigms in resource management; it adds another dimension by solving NP-hard optimization problems in resource scheduling with unprecedented efficiency. Quantum cloud platforms allow users to access quantum resources without needing specialized hardware [41]. It enables advanced simulation models for demand prediction and introduces quantum-inspired approaches for near-optimal resource management. Efficient scheduling and resource management are essential to handle the unique characteristics of quantum computations, such as qubit fidelity and queuing times.

5 Current state of resource management approaches

Allocation strategies and scheduling methods in cloud computing utilize a variety of approaches to meet consumer requirements effectively [45]. As the number of user requests increases, efficient scheduling becomes crucial; as poor scheduling can lead to significant performance degradation. An effective task scheduler must be capable of adapting to diverse scenarios and task types [46,47]. Existing optimization algorithms often address optimization problems and the inherent complexity of the optimization problems but often prioritize exploitation over exploration, limiting their ability to discover optimal solutions [48,49]. This section examines the performance of optimization algorithms, including mathematical, heuristic, and hyper approaches. With a particular emphasis on the advantages and limitations of MHs. Figure 5 illustrates a new taxonomy of methods categorizing them based on their underlying approaches.

5.1 Mathematical approach

The mathematical approach aims to find optimal solutions or near-optimal solutions by formulating the problem as a mathematical model and solving it using optimization techniques, but they may require more

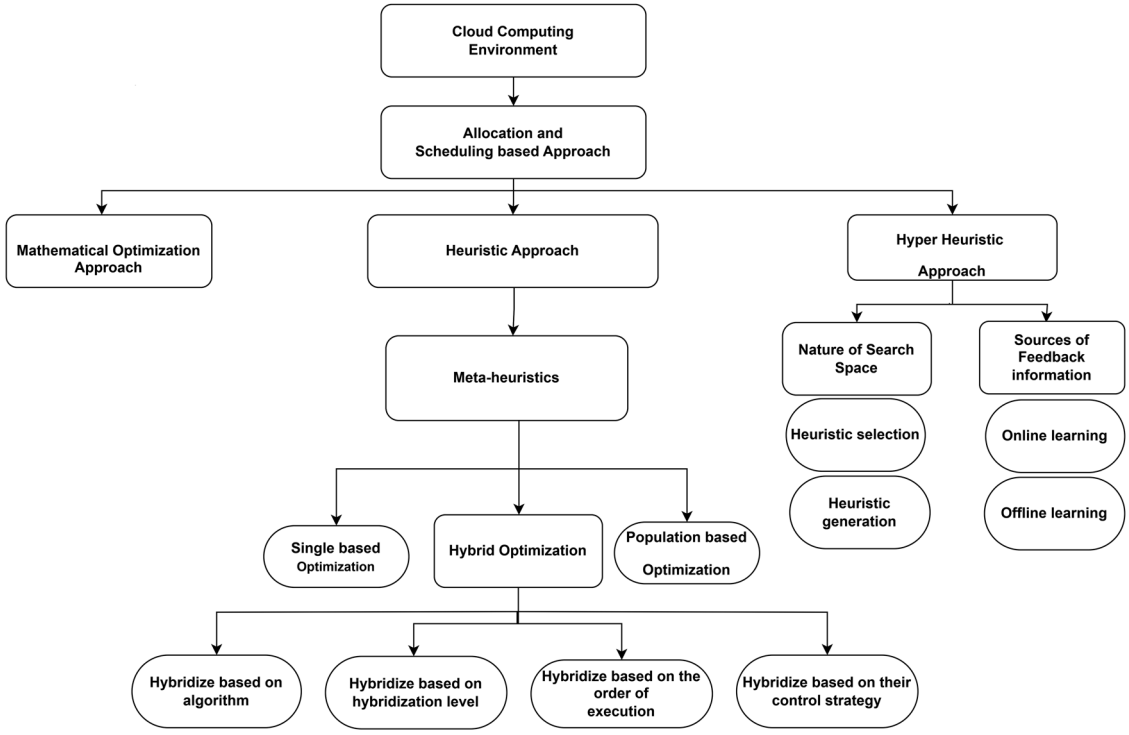

Figure 5: A new taxonomy diagram representing categorization of approaches based on their methodologies (created by the authors).

computational resources and time [50]. The most popular examples are integer linear programming (ILP) [51], nonlinear programming [51], dynamic programming [52], and game theory-based approaches [52].

These techniques have been applied in various fields, such as AI, operations research, engineering, and astronomy. This is due to their ability to provide structured solutions to many complex problems [53]. For example, ILP is effective in scenarios where decisions are discrete, such as in resource allocation and scheduling problems. On the other hand, when the relationships between variables are not linear, then nonlinear programming is used [54].

While dynamic programming is particularly used in scenarios that involve sequential decision-making, such as inventory management and shortest path problems, it is effective for solving problems that can be broken down into simpler subproblems. Game theory-based methods provide insights into competitive situations where the outcome depends on the actions of multiple agents, and thus provide strategic frameworks for negotiations, pricing strategies, and conflict resolution [51]. To strike a balance between computing efficiency and optimality for addressing real-world issues, researchers continue to explore hybrid approaches that blend heuristic strategies with mathematical rigor [55].

5.1.1 Review of mathematical approach

This section reviews articles on resource allocation and task scheduling algorithms that are based on mathematical approach; Table 2 lists the algorithms used in cloud computing environments.

Zhu et al. [51] suggested a dynamic pricing system based on a model for the Stackelberg game to tackle the challenge of increasing the income for cloud computing providers of Infrastructure as a Service (IaaS) and Software as a Service (SaaS). The simulation results show that the suggested mechanism performs better than auction-based pricing and fixed pricing systems in terms of revenue maximization and resource usage.

A multi-resource fair scheduling (MRFS) algorithm for heterogeneous cloud computing environments was presented by Hamzeh et al. [52]. The study aims to maximize each user's utility on a specific server by reducing

Table 2: Mathematical algorithms used in resource allocating and task scheduling

Algorithms/ approaches	Simulation/ tool	Objective(s)/key performance metrics	Quantitative results	Sample size (no. of cloud nodes, tasks, or datasets	Main focus area	Limitations
Stackelberg game model for dynamic pricing [51]	CloudSim	Maximizes revenue and resource utilization	Surpassed fixed and auction-based pricing in resource utilization and revenue	Three SaaS cloud providers and one IaaS provider, with three service types, and one type of VM instance	Resource allocation	The study runs under a simplified scenario and does not accurately capture the complexity of real-world cloud computing infrastructures
MRFS [52]	CloudSim	Utility maximization, resource dominance minimization	Improved solution quality, maximized resource use, and reduced server count	Three users distributed across three servers, each with a varying VM request profile	Resource allocation	Specific data used not specified
ILP [57]	MATLAB	Task completion time and payoff level	The proposed method outperformed RR method in task completion time and reward	IoTcloudServe@TEIN platform (two cloud scenarios)	Task scheduling	Time-series structure of requests not addressed
Enhanced ordinal optimization with linear regression [58]	CloudSim	Makespan minimization	Reached the target minimum makespan, and narrowed the search space	Not specified	Task scheduling	No comparison with existing approaches
Lagrangian relaxation, Mathematical programming [59]	Not specified	Energy consumption, task execution time, and resource utilization	Reduce energy operations and achieved high performance	Not specified	Resource allocation and Task scheduling	Trade-offs between energy efficiency and additional metrics not addressed
DCRNN [60]	CloudSim	Error of root-mean-square, Error of average absolute percentage	Root mean square error: 2.40%, Mean absolute percentage error: 0.18%	PlanetLab CPU usage data	Resource allocation	Focus only on CPU utilization, ignoring other metrics
Linear programming (LP-WSC) [61]	Not specified	QoS parameters, cost, availability, reliability	The proposed method outperformed in terms of significant cost reduction and improved reliability compared to traditional methods	Amazon EC2 big dataset taken from personal cloud datasets	Resource allocation	No comparison with state-of-the-art methods
BB-BC model [62]	CloudSim	Cost and time	Exceeded the performance of bio-inspired, static, dynamic methods	Not specified	Resource allocation	Limitations not discussed, data source not mentioned
Mixed set programming [63]	CloudSim	Resource allocation efficiency, collaboration efficiency	Improved efficiency in inter-enterprise collaboration	Not specified	Resource allocation	No comparison with existing approaches, no large-scale experiments

the number of dominant resources. MRFS was introduced as a solution to the resource scheduling problem in cloud computing. It considers the fair and efficient allocation of resources to users. However, the study did not specify the specific data used. It focuses on introducing the MRFS algorithm and assessing its effectiveness.

Researchers presented a method for allocating resources to meet growing needs in cloud computing [56]. The study introduced a multi-objective optimization technique aimed at aligning resource performance with percentage distances and minimizing the number of physical servers utilized. The RAA-PI-NSGAII method not only maximizes resource utilization and reduces solving time but also improves the quality and uniformity of the solution set distribution. However, the study does not adequately address the specific challenges or limitations encountered during the implementation of the proposed resource allocation algorithm, particularly in the context of increasing cloud computing demands.

A mathematical model to enhance the performance and efficiency of cloud computing by scheduling tasks within containers was introduced by Swatthong and Aswakul [57]. The model employs ILP to increase the average compensation of tasks while adhering to resource and demand constraints. This study evaluated the model using two cloud scenarios: An edge-core cluster and a peer-to-peer federated cloud. Results demonstrated that the proposed model outperformed the standard RR scheduling method in terms of task completion time and reward level. Additionally, the study underscores the importance of flexible and fine-grained task separation in cloud architecture. However, it does not account for the time-series nature of the requests, which could impact the model's ability to handle maximum load efficiently.

Yadav and Mishra [58] suggested an enhanced ordinal optimization method paired with linear regression to enhance scheduling of task in cloud computing to reduce the makespan. This method narrows down the search space for scheduling and efficiently assigns the workload to the best schedule. Additionally, it forecasts future dynamic workload to achieve a target minimum makespan. However, there is no comparison with existing approaches to validate the effectiveness of the proposed technique.

Tai et al. [59] proposed an enhanced algorithm for managing computing resources and energy consumption in heterogeneous cloud computing centers. The algorithm considered factors such as energy usage, task scheduling, execution time, employing Lagrangian relaxation and mathematical programming, which develop high-performance and energy-efficient cloud computing facilities. However, the study does not address the theoretical trade-offs between energy efficiency and other performance metrics, such as task execution time or resource utilization.

Al-Asaly et al. [60] presented a self-sufficient, intelligent workload prediction technique employing a diffusion convolutional recurrent neural network (DCRNN) model for cloud resource allocation. The objective is to enhance prediction precision and reduce the gap between forecasted and real workloads in cloud computing setups. Tested on real PlanetLab CPU usage data, the model surpassed other deep learning models, achieving a root-mean-square error of 2.40% and an average absolute percentage error of 0.18%. However, the study focuses specifically on CPU utilization as the input data for prediction model; it ignores other metrics or factors such as memory consumption or network traffic.

Ghobaei-Arani and Souri [61] introduced a linear programming method called LP-WSC for web service composition in geographically distributed cloud environments, intending to improve QoS parameters. This approach selects the most efficient service for each request and significantly reduces the cost of choosing and configuring services, while increasing the availability of services and reliability of servers compared to other methods. However, the suggested LP-WSC strategy is not compared to other cutting-edge methods for web service orchestration in geographically dispersed cloud environments in this study.

In order to provide variable job assignments on VMs, Rawat et al. [62] proposed a cost-effective model of Big-Bang Big-Crunch (BB-BC) for resource allocation in cloud setups. It targets a globally optimal outcome through an objective function that considers metrics such as cost and time, surpassing traditional static, dynamic, and bio-inspired provisioning methods. However, the authors did not discuss the limitations and the data source was not mentioned in the study.

Shi et al. [63] proposed a uniform model description of manufacturing resources using ontology and metadata modeling methods. They also outline a strategy for collaborative scheduling of manufacturing resources between enterprises using Mixed Set Programming, which enhances resource collaboration and allocation efficiency. In subsequent work, the authors suggest a model and scheduling mechanism that can be

integrated with the study of inter-enterprise logistics to improve the efficiency of inter-enterprise resource collaboration further. However, the study does not compare the proposed model and scheduling strategy with the existing approaches for conducting large-scale experiments.

5.2 Heuristic approach

The heuristic approach depends on empirical rules, guidelines, or experience-based decisions to assign resources and schedule workloads based on simple criteria without considering global optimization [64]. These methods remain popular for their computational efficiency and capacity to offer nearly optimal solutions. The two main parts of the heuristic-based techniques are specific heuristics and MHs. The most famous examples include first come first serve (FCFS) [65], shortest remaining time [65], RR [66], GAs [67], and PSO [67,68].

The heuristics are particularly useful in some scenarios and ineffective in others under heavy loads. These strategies take advantage of problem-specific knowledge to simplify complex decision-making processes, enabling faster responses in dynamic environments. For example, the “first come, first served” principle is commonly used in task scheduling, where the order of execution is determined by the arrival sequence. This makes it simple, but can sometimes be inefficient when dealing with large loads [64]. On the other hand, techniques such as RR are used for distribution of resources by allocating fixed time slots to each task fairly, which is especially useful in multi-user systems. This method mitigates the risk of starvation, ensuring that all processes get attention within a reasonable time frame [69].

In contrast, MH techniques, such as GAs and PSO, introduce a higher level of complexity by exploring a larger solution space. These techniques iteratively improve solutions by utilizing mechanisms inspired by natural processes, such as social behavior in PSO or selection, crossover, and mutation in GAs [70].

Notwithstanding their advantages, heuristic-based approaches may produce suboptimal solutions, especially when the underlying assumptions do not hold true in practice. Therefore, even though they provide valuable insights and speedy fixes, these methods must be used in conjunction with more exacting optimization strategies when accuracy is crucial. As research continues to evolve, hybrid models that combine heuristics with exact algorithms are gaining traction, promising to harness the strengths of both paradigms to achieve superior outcomes in resource allocation and workload scheduling [43].

5.2.1 MH approach

MH are strategies that provide efficient and optimized solutions, helping to solve problems of complex optimization in various fields by providing acceptable answers in a reasonable amount of time. MH search techniques are general, sophisticated approaches that can be used to build basic heuristics to solve specific optimization problems [71]. Since scheduling is an NP-hard problem, most scheduling algorithms do not explore the entire problem space to find the optimal resource allocation; hence, MHs are the best choice for this problem [72].

(a) Single-based optimization algorithms: Generate a single random solution and strive to enhance it through optimization processes [73].
(b) Population-based optimization algorithms (POAs): POAs rely heavily on factors such as the choice of algorithms, strategies, parameter combinations, constraint handling methods, local search methods, surrogate models, and niching methods to determine solution quality for optimization problems, etc. [73].
(c) Hybrid MH: A hybrid MH primarily distinguishes according to four criteria [74]:

- Hybridization based on algorithm: Hybridized algorithms include combinations such as MHs with MHs, MHs with Heuristics, MHs with Fuzzy logic techniques, or MHs with statistical techniques.
- Hybridization based on hybridization level: Combine algorithms based on their level of coupling strength.

- Hybridization based on the order of execution: The individual algorithms are executed either sequentially, intertwined, or even in parallel.
- Hybridization based on their control strategy: Hybrids are classified based on their control strategy, which can be either integrative (coercive) or collaborative (cooperative).

5.2.2 Review of heuristic and MH algorithms

This subsection reviews articles on resource allocation and task scheduling algorithms based on heuristic and MH algorithms, including single-based, population-based, and hybrid algorithms. Table 3 lists the algorithms used in cloud computing environments.

Chhabra et al. [29] addressed the optimization of bag-of-tasks scheduling in cloud data centers by developing the opposition learning enabled whale particle swarm optimization (OWPSO) algorithm, which combines the whale optimization algorithm (WOA) with PSO to improve scheduling efficiency. The study evaluates key performance metrics, including makespan and energy consumption, demonstrating that OWPSO significantly outperforms baseline algorithms in producing near-optimal scheduling solutions. However, the reliance on synthetic datasets for benchmarking and the need for thorough parameter tuning to achieve optimal performance limit the practical applicability of the suggested technique.

Mirmohseni et al. [39] presented the fuzzy PSO (FPSO)-GA approach, which combines GA and fuzzy PSO techniques to achieve effective load balancing in cloud networks. The proposed approach demonstrates significant improvements in load balancing and reducing energy, surpassing algorithms such as LBPSGORA, PSO, and GA in load balancing performance. While the authors highlight the effectiveness of the FPSO-GA algorithm in enhancing load distribution, the study lacks detailed numerical results or specific performance metrics to substantiate the claimed improvements.

Manavi et al. [68] proposed a novel hybrid approach for resource allocation and task scheduling in cloud computing, combining neural network classification with GA optimization. The method aims to improve execution time, cost, response time, and system utilization while considering fairness to prevent task starvation. Using a large Google dataset for simulation, the authors demonstrate improvements of 3.2% in execution time, 13.3% in cost, and 12.1% in response time compared to state-of-the-art methods. However, the study is limited to independent tasks without deadlines and uses a relatively small chromosome size in the GA, which may impact scalability for larger task sets.

Zhu et al. [70] proposed a heuristic multi-objective task scheduling framework for container-based clouds, exploiting actor-critic RIL to enhance task scheduling efficiency. The framework addresses the complexities of resource allocation and task management in dynamic environments, showing significant improvements in metrics such as resource utilization and execution time through extensive simulations. However, the study's applicability is constrained by its focus on specific cloud environments, potentially limiting the generalizability of the findings. Additionally, the framework may face challenges with real-time adaptability in highly variable workloads.

Ibrahim et al. [75] presented a comparative evaluation of state-of-the-art static task scheduling algorithms in cloud computing, focusing on their performance in terms of resource utilization, makespan, throughput, and response time. The study compares methods such as PSSLB, MCT, min-min, max-avg, LBIMM, RASA, Sufferage, and TASA using CloudSim simulations with HCSP and GOCJ datasets. The results show that while MCT and Sufferage perform well in reaction time but poorly in resource utilization, TASA and PSSLB perform better across various metrics and datasets. The study emphasizes how crucial it is for cloud service providers and end users to balance resource usage and execution time. Nevertheless, the study is constrained by its emphasis on static scheduling techniques and dependence on virtualized settings instead of actual cloud configurations.

Hamid et al. [76] compared different task scheduling algorithms of cloud computing based on makespan using workflows as datasets, with makespan serving as the primary performance metric. Experimental results indicate that the FCFS algorithm outperforms RR, min-min, and max-min in the CyberShake workflow, while max-min outperforms FCFS, RR, and min-min in the Montage workflow. However, the study does not explore

Table 3: Heuristic and MH algorithms used in resource allocating and task scheduling

Algorithms/ approaches	Simulation/tool	Objective(s)/key performance metrics	Quantitative results	Sample size (no. of cloud nodes, tasks, or datasets)	Main focus area	Limitations
OWPSO [29]	CloudSim	Makespan reduction, and reduce energy consumption	Near-optimal solutions, and enhanced improvements in scheduling efficiency	4,800 scheduling experiments on CEA-Curie and HPC2N workload	Task scheduling	Insufficient exploration-exploitation phase trade-off inadequate exploration ability, high computation complexity, and slow convergence
N2TC, and GATA [68]	Simulation-based experiment	System utilization, cost, response time, and execution time	3.2% in execution time, 13.3% in cost, 12.1% in response time	405,894 tasks from Google cluster-traces v3 dataset	Task scheduling, and resource allocation	Limited to independent, non-preemptive tasks without deadlines
AC-CCTS (Actor-Critic Container Task Scheduling) [70]	CloudSim	RUR, RBD, and QoS	Improved utilization and reduced execution time via extensive simulations	Not specified	Resource allocation and task scheduling	Scheduling complexity due to dynamic workloads and environmental variability
LBIMM, Max-Avg, MCT, min-min, PSSLB, RASA, Sufferage, TASA [75]	CloudSim	ARUR, makespan, Throughput, ART	PSSLB and TASA perform well across metrics	HCSP dataset (four instances), GOCJ dataset	Task Scheduling	Restricted to static scheduling algorithms
FCFS, min-min, max-min, and RR [76]	WorkflowSim	Makespan	Max-min is best in Montage workflow; FCFS excels in CyberShake	CyberShake and Montage workflows datasets	Task scheduling	Makespan measurement not discussed for increased data centers/hosts
Improved backfilling algorithm [77]	MATLAB	Task acceptance ratio, task rejection ratio	91.94% acceptance ratio, 8.05% rejection ratio	Open Nebula cloud platform	Task scheduling	Limited number of workloads and parameters used
Deep RIL [78]	CloudSim	Energy consumption, makespan, and SLA violation	Dynamic optimization of task execution	HPC2N, and NASA worklogs	Task Scheduling	Lack of detailed discussion on algorithm limitations
Dynamic resource-aware load balancing algorithm (DRALBA), Resource-aware load balancing algorithm (RALBA), Dynamic load balancing algorithm (DLBA), min-min, max-min, RR [79]	CloudSim	Makespan, ARUR, and Throughput	DRALBA excels competitors in both synthetic and realistic workloads	Synthetic and realistic workloads	Task scheduling	Scalability and real-world performance not discussed
TDSA [80]	Not specified	Makespan and resource utilization	17.4% improvement in makespan, 31.6% in utilization	Randomly generated and real workflows	Task scheduling	Running time and data transfer uncertainty not considered
Multiclass priority task scheduling (MCPTS), DE ELECTRE III [12]	CloudSim	Task priority, queueing priority, resource priority	Dynamic task priority adjustment enhances efficiency	(KTH) IBM SP2 log	Task scheduling	Insufficient consideration of dynamic request characteristics

(*Continued*)

Table 3: *Continued*

Algorithms/ approaches	Simulation/tool	Objective(s)/key performance metrics	Quantitative results	Sample size (no. of cloud nodes, tasks, or datasets)	Main focus area	Limitations
DRRHA [81]	CloudSim	Turnaround time, average waiting time and response time	Enhanced scheduling efficiency	Real dataset and a randomly generated dataset	Task scheduling	No comparison with state-of-the-art algorithms for resource usage or energy efficiency
Min-min and Density-based spatial clustering of applications with noise (DBSCAN) [82]	CloudSim	Execution time, task completion rates, and defect reduction	Enhanced task completion rates and reduced defects	NASA iPSC workload log file	Task scheduling	Complex job scheduling procedure, limited comparison with existing algorithms
LJFP-PSO and MCT-PSO [84]	MATLAB	Makespan, execution time, degree of imbalance, and total energy consumption	Enhanced performance compared to traditional PSO	Not specified	Task scheduling	Specific dataset or real-world scenarios not detailed
MTWO [85]	CloudSim	Makespan, throughput, response time, degree of imbalance, power efficiency, and resource utilization	Improved task scheduling and resource allocation	Not specified	Task scheduling, and resource allocation,	Specific data used in simulation not mentioned
GWO with RIL [86]	CloudSim	Runtime	Substantial decrease in uptime	Not specified	Task allocation	Assumes prior knowledge of task computational time
ICOATS [88]	CloudSim	Makespan and resource utilization	Improved resource utilization and makespan	Real-time worklogs derived from NASA	Task scheduling	Limited consideration of QoS characteristics
IWC [89]	MATLAB	Convergence speed, and accuracy	Faster and more accurate convergence	Small- and large-scale computing	Task scheduling	Parallel applications not investigated
Q-ACOA (improved Ant Colony Optimization) [90]	CloudSim	Task completion time, data migration time, and cost	Exceeds alternative algorithms	Not specified	Resource allocation, and task scheduling	Dataset not specified
ANN-BPSO [91]	CloudSim	Resource utilization, and reduce response times	Enhanced scheduling and load valancing in cloud environments	Not specified	Resource allocation, and task scheduling	Dataset not specified
HCSOA-TS [92]	CloudSim	Makespan, execution cost, and load balance	Outperforms GA, PSO, and GA-PSO	Four scenarios tested: (1) 25 tasks, 95 edges, (2) 50 tasks, 206 edges, (3) 100 tasks, 433 edges, (4) 1,000 tasks, 4,485 edges	Task scheduling	The study does not consider real-world cloud workloads or infrastructures
BSO-LB [93]	CloudSim	Utilizes resources and reduces makespan	Outperforms load balancing	GoCJ: Google cloud jobs dataset	Task scheduling	The proposed algorithm is slower than other algorithms

(*Continued*)

Table 3: *Continued*

Algorithms/ approaches	Simulation/tool	Objective(s)/key performance metrics	Quantitative results	Sample size (no. of cloud nodes, tasks, or datasets)	Main focus area	Limitations
EHJSO [94]	CloudSim	Makespan, computation time, fitness, iteration-based performance, success rate	Outperforms previous methods	Not specified	Task scheduling	The authors did not specify limitations of the proposed algorithm
Reference Vector Guided Evolutionary Algorithm (RVEA-NDAPD) [96]	CloudSim	Makespan, execution cost, VM load, and user expectations	Enhanced performance over MaOEAs	Not specified	Task scheduling	Lack of detailed data information
Total resource execution time aware algorithm (TRETA) [97]	CloudSim	Total execution time, and degree of imbalance	Improved resource usage and performance	Real-world workload traces of NASA Ames iPSC/860	Task scheduling	Not evaluated with other tactics or measures
AMO-TLBO [100]	WorkflowSim	Utilization, space, and cost	Outperforms TLBO, MOPSO, and NSGA-II	Not specified	Resource allocation	Dataset source not specified
GAECS [105]	MATLAB	Makespan, energy consumption, load balancing, and task completion time	Enhanced allocation and reduced makespan	Stochastic Datasets	Task scheduling	Complex parameter tuning required

how makespan is affected as the number of data centers or hosts increases, leaving a critical aspect of scalability unaddressed.

Nayak et al. [77] intended to minimize the task rejection ratio and maximize the task acceptance ratio in a cloud-based scheduling mechanism for activities with deadlines. It improves upon the existing backfilling algorithm by addressing conflicts among similar leases and allowing the scheduling of new deadline-sensitive tasks during execution. An average lease acceptance ratio of 91.94% and a minimal average lease rejection ratio of 8.05% are attained by the suggested mechanism. It considers the current time and gap time as scheduling parameters, that are not addressed in the current studies. It allows the arrival of a new lease to be planned and does not require a decision maker such as analytic hierarchy process to resolve conflicts between similar leases. However, the performance analysis is based on limited workloads and parameters, which may not fully represent the diverse requirements of different cloud applications.

Mangalampalli et al. [78] proposed a deep RIL-based task scheduling algorithm in cloud computing, designed to enhance significant QoS parameters such as SLA violation, energy consumption, and time. The algorithm dynamically calculates task execution time based on a threshold value and the load on physical hosts, with host utilization determined using specific equations. While the approach effectively addresses key parameters such as makespan, SLA violations, and energy consumption, the study lacks a comprehensive discussion of the algorithm's limitations. Including such insights could have highlighted potential drawbacks and opportunities for future refinement.

Mishra and Gupta [79] presented a study that compares heuristic algorithms for scheduling tasks in cloud computing, such as RALBA, DRALBA, DLBA, max-min, min-min, and RR, based on performance parameters such as throughput, makespan, and average resource utilization ratio (ARUR). The existing DRALBA approach outperforms other approaches in terms of performance parameters, both on realistic workloads and synthetic, making it an efficient and effective scheduling algorithm for cloud computing environments. However, there is no discussion regarding the suggested algorithms' scalability and performance with larger workloads or in real-world cloud computing environments.

Yao et al. [80] proposed a task duplication-based scheduling algorithm (TDSA) to enhance the makespan for budget-limited workflows, utilizing idle slots on resources and reallocating the leftover budget. The TDSA algorithm shows notable enhancements in the makespan of workflows (up to 17.4%) and the utilization of cloud computing resources (up to 31.6%) compared to the four baseline algorithms, as evidenced by experiments on randomly generated and actual workflows. However, the study does not consider the running time of workflow tasks, and the amount of data transferred among workflow tasks is highly uncertain.

Ben Alla et al. [12] proposed a novel approach to address the problem of user requests and supplier resources not being prioritized. They proposed an efficient priority task scheduling scheme called MCPTS, in which four task parameters (length, waiting time, deadline, and burst time) are used to determine priority. The task priority, task queuing priority, and resource priority sub models make up the MCPTS scheme. To determine the priority of activities, they suggest using differential evolution (DE), a MH algorithm, and elimination and choice expressing reality version III, a new hybrid multi-criteria decision-making method. They also presented a queueing model-based dynamic priority-queue algorithm. Moreover, they created a productive and adaptable relationship between resource and task classes by dynamically adjusting resource priority depending on the task's priority model. However, insufficient consideration of dynamic request characteristics and resource availability could affect scheduling choices.

To schedule tasks in cloud computing systems, Alhaidari and Balharith [81] proposed a novel method known as the dynamic RR heuristic algorithm (DRRHA). This method performs noticeably better than previous algorithms evaluated in terms of average waiting time, turnaround time, and response time. DRRHA employs the RR algorithm to increase task scheduling efficiency, modifying its time quantum based on the task's remaining burst time and time quantum means. However, there has been no comparison between the proposed algorithm and other state-of-the-art task scheduling algorithms regarding resource usage or energy efficiency.

Mustapha and Gupta [82] introduced a fault-aware task scheduling algorithm in cloud computing using min-min and DBSCAN to enhance resource allocation efficiency fault tolerance, and enhance QoS in a dynamic and heterogeneous environment. The approach outperforms existing algorithms such as ant colony

optimization (ACO), PSO, BB-BC, and whale harmony optimization (WHO) in terms of execution time, task completion rates, and defect reduction. However, the proposed technique uses DBSCAN to cluster resources, complicating the job scheduling procedure. Moreover, comparing the suggested algorithm with fewer existing algorithms may not provide as clear a picture of the algorithm's performance as a more extensive comparison with a wider range of state-of-the-art methods.

Khan [83] released a power-efficient cloudlet scheduling (PACS) approach to reduce energy consumption, request processing time, and cloud computing setup costs. When compared to other popular cloudlet scheduling methods, PACS provides a noteworthy 3.80–23.82 speed boost. However, the study did not evaluate how well the suggested PACS technique scales in handling higher numbers of cloudlets and VMs.

Alsaidy et al. [84] reported that efficient task scheduling in cloud computing was vital for cost-effective execution and resource utilization, as addressed by the LJFP-PSO and MCT-PSO algorithms. These algorithms employ heuristic initialization to boost PSO performance, surpassing traditional PSO and comparative techniques in reducing makespan, execution time, imbalance, and energy consumption metrics. However, the study does not provide details about the specific dataset or real-world scenarios for evaluating the proposed algorithms.

Prathiba and Sankar [85] presented the multi-task wolf optimizer (MTWO) method, merging efficient task scheduling and secure resource allocation in cloud computing to tackle data and resource management challenges in a rapidly expanding cloud setting. This technique showcases a brief task schedule employing wolf optimization techniques to minimize makespan time and boost throughput. It further incorporates a deep neural network with cluster optimization for resource allocation efficiency within architectural limits, enhancing response time, power efficiency, and resource utilization. However, the study does not explicitly mention the specific data used in the simulation setup and analysis.

Yuvaraj et al. [86] introduced a machine-learning technique to optimize job allocation between the contributor and the event queue in the serverless framework. To increase the effectiveness of work distribution, they applied the gray wolf optimization (GWO) model. Moreover, the authors optimized GWO settings and improved work allocation using an RIL technique. According to simulation experiments, the suggested GWO-RIL technique significantly reduces runtimes and adapts to shifting load situations. However, the study assumes prior knowledge of the computational time period for each task and similar overheads prior to task scheduling, which may not always be realistic in real-world scenarios.

Nanjappan et al. [87] proposed a technique that integrates adaptive neuro fuzzy inference system (ANFIS) and black widow optimization (BWO) to enhance resource utilization and scheduling in a cloud computing. The ANFIS-BWO method determined which VM was best suited for each task. The BWO technique identifies the optimal solution for the ANFIS scheme. The proposed approach aims to minimize computation time, cost, and energy consumption while optimizing resource utilization. Nevertheless, it was not made clear which particular datasets were used in this investigation.

Tamilarasu and Singaravel [88] proposed the Coati Optimization Algorithm-Based Task Scheduling (ICOATS) to tackle challenges related to the scheduling of task in cloud environment. ICOATS aims to tackle issues such as long scheduling times, increased costs, and optimized VM workloads. A multi-objective fitness function is seamlessly integrated. This function aims to simultaneously reduce the makespan while enhancing the utilization of available resources. An exploitation strategy is incorporated to achieve this, which helps avoid premature convergence and improves the local search potential. ICOATS aims to find efficient solutions for cloud task scheduling by striking a harmony between exploitation and exploration. However, the study's effectiveness in achieving optimal solutions may be limited due to the lack of consideration for a comprehensive set of QoS characteristics during the optimization process.

An improved WOA algorithm for cloud task scheduling (improved whale optimization [IWC]), using the WOA was described by the researchers [89] to improve task scheduling in cloud computing systems. Optimizing task scheduling plans and resource utilization to increase cloud system performance is one of the most important functions of IWC technology. The proposed IWC algorithm shows superior convergence speed and accuracy when compared to existing MH algorithms. It is versatile and can be applied to both small- and large-scale problems. However, the authors did not investigate parallel applications in cloud environments to reduce the scheduling overhead of this method when dealing with large workloads.

The researchers described an improved version of the ant colony optimization algorithm (ACOA) called Q-ACOAn [90] designed to fulfill predetermined time and cost objectives. This aims to enhance the algorithm for allocating resources and scheduling tasks in cloud computing. Results demonstrate that Q-ACOA outperforms alternative scheduling algorithms in job completion time, data migration time, and overall cost efficiency. Nonetheless, the study lacked explicit information regarding the dataset used in the research.

Alghamdi [91] introduced a novel resource allocation method for cloud computing environments using binary PSO (BPSO) and artificial neural networks (ANN). The proposed method aims to improve particle placements and thus reduce work completion times across VMs. Reducing response times, improving resource utilization, and ensuring good QoS for cloud computing applications are the project's primary goals. Researchers worked to improve load and scheduling of the task in cloud environments using the proposed ANN-BPSO approach.

Shrichandran et al. [92] introduced a hybrid competitive swarm optimization algorithm for task scheduling (HCSOA-TS) in cloud environments. The method incorporates a Cauchy mutation operator into the Competitive Swarm Optimization algorithm to improve performance. The researchers assessed the HCSOA-TS technique across four situations with different numbers of tasks and edges, comparing it to GA, PSO, and GA-PSO algorithms. Performance metrics included makespan, execution cost, and load balance. Results showed that HCSOA-TS outperformed the other algorithms across all scenarios. However, the study is limited by its simulation-based nature, relatively small-scale scenarios (maximum 1,000 tasks), and lack of consideration for real-world cloud workloads.

The bird swarm optimization load balancing (BSO-LB) algorithm was the load balancing method suggested by Mishra and Majhi [93]. The algorithm views VMs as destination food patches and tasks as birds. The suggested approach seeks to reduce response time and optimize load balancing in cloud system. The load balancing technique is paired with the binary variation of the BSO algorithm. Experimental findings demonstrate that the suggested approach surpasses alternative algorithms. However, it is not as fast as other algorithms.

Paulraj et al. [94] covered the significance of task scheduling in cloud computing and suggested an effective hybrid job scheduling optimization method called efficient hybrid job scheduling optimization (EHJSO), which combines Cuckoo Search Optimization and GWO. Metrics such as makespan, computation time, fitness, iteration-based performance, and success rate are used to compare the suggested method to prior research and are determined to be superior. However, the authors do not specify the limitations of the proposed algorithm.

Hu et al. [95] presented well-organized scheduling algorithm based on energy for processing real-time applications in cloud computing, aiming to abate energy consumption while meeting real-time requirements. The method demonstrates an important decrease in energy consumption compared to existing algorithms, with a higher success rate in finding feasible schedules and comparable computation time. However, the study lacks specific details about the nature or source of the real-case benchmarks or the data used in the synthetic application.

Xu et al. [96] presented a reference vector-guided evolutionary algorithm (RVEA-NDAPD) and a many-objective scheduling strategy for cloud environments in order to solve the model. The performance of the suggested model in a cloud computing environment was effectively improved by the RVEA-NDAPD algorithm compared to the traditional many-objective evolutionary algorithms (MaOEAs) currently in use. However, the study lacks detailed information on the specific data used in the study, including task characteristics, system and user details, as well as performance parameters of VMs and tasks.

Bandaranayake et al. [97] proposed a TRETA method for cloud task scheduling to optimize the total execution time of computing resources. The study demonstrates that the suggested method enhances performance and resource usage for cloud jobs by comparing it with alternative heuristics on real-world workloads. The authors did not evaluate the algorithm when combined with other tactics or in relation to other measures.

Alsubai et al. [98] proposed an innovative swarm-based task scheduling approach that integrates the moth swarm algorithm and the chameleon swarm algorithm to optimize cloud scheduling in terms of efficiency and security. The objective is to enhance the tasks distribution using available resources and encode cloud information during task scheduling, with a focus on optimizing the available bandwidth for efficient

scheduling of cloud computing tasks. Improvements in measures such as imbalance score, throughput, cost, average waiting time, reaction time, throughput, latency, execution time, speed, and bandwidth utilization are demonstrated by evaluating the performance of the approach. However, the specific details of the proposed algorithm and its technical implementation are not provided in the sources.

Kaur and Kaur [99] suggested a hybrid approach to improve load balancing in cloud systems by combining heuristic and MH methods. Within the HDD-PLB framework, two hybrid strategies were proposed and examined: hybrid heterogeneous early finishing time (HEFT) with ACO (hyper-heuristic algorithm [HHA]) and hybrid prediction early finishing time with ACO method (HPA). The authors assessed the efficiency of the suggested architecture using two key performance metrics: manufacturing scale and cost. Although the authors claim that the framework operates optimally in terms of cost and scale, no experimental evaluation or findings were offered to support their assertions.

Moazeni et al. [100] presented a new approach for cloud computing resource allocation, relying on the multi-objective-learning-based-optimization (AMO-TLBO). The AMO-TLBO algorithm includes features including the number of teachers, teaching factor adaptive, and self-motivated instructional learning to improve the skills of exploration and exploitation. Increasing utilization while reducing space and cost is one of the most important features of the proposed AMO-TLBO algorithm. Evaluation results show that the proposed method outperforms the TLBO, MOPSO, and NSGA-II algorithms in several performance metrics. However, the source of the dataset is not specified.

Zhao et al. [101] proposed an innovative cloud resource allocation strategy that prioritizes requests into two categories: high-priority primary requests (PRs) and low-priority secondary requests (SRs). This approach implements an entry threshold and probability mechanism to regulate SR admission, thereby enhancing service quality for PRs. The team developed a discrete-time queuing model to evaluate the strategy's effectiveness and calculated various performance metrics. Experimental results were promising, showing a significant reduction in PR blocking rate by 47% and an impressive increase in PR throughput rate of up to 55% when compared to traditional strategies without entry control. However, the study has a notable limitation: it lacks comparative analysis with other contemporary resource allocation strategies. This omission makes it difficult to fully assess the relative performance and effectiveness of the proposed approach within the broader context of cloud resource management solutions.

Aktan and Bulut [102] discussed the advancement of MH and hybrid MH algorithms for task scheduling in cloud computing set to enhance completion time and load balancing of VMs. They evaluate various algorithms such as GA, DE, simulated annealing (SA), and their hybrid versions based on completion time and load balancing. The results indicate that the hybrid DE-SA algorithm outperforms the single DE and SA algorithms in completion time. Moreover, it improves the average completion time and standard deviation for specific task sets of VM loads. However, the study does not compare the proposed algorithms with existing state-of-the-art methods.

Albert and Nanjappan [103] presented a cloud task scheduling strategy using a hybrid WHO algorithm to balance the system load, minimize makespan, and reduce costs. Experimental results show that the algorithm achieves high load balance with decreased execution time, cost, and energy consumption, proving its effectiveness in cloud resource management. The proposed WHOA algorithm outperforms other algorithms such as OGWO, WOA, and HS in terms of cost efficiency and makespan optimization. However, there is no discussion of the challenges that may arise when implementing the proposed algorithm in a real-world cloud computing environment.

Nekooei-Joghdani and Safi-Esfahani [104] presented a new hybrid MH algorithm named Gabor opposition-based learning multi-verse optimizer for a dynamic cloud environment to solve the scheduling problem, demonstrating superior performance compared to baseline algorithms. GOMVO combines Gabor filter, multi-verse optimizer, opposition-based learning, and multi-tracker optimization algorithms to improve task allocation in cloud environments. The multi-verse optimizer-Gabor opposition-based learning multi-verse optimizer (MTOA-GOMVO) hybrid algorithm further enhances the resolution of premature convergence issues by leveraging the strengths of both the MTOA and GOMVO algorithms. While the proposed MTOA-GOMVO hybrid algorithm shows promising results in improving task scheduling in cloud computing environments, its scalability to larger and more complex scenarios could be a concern.

Pirozmand et al. [105] explored the application of a multi-objective hybrid GA for cloud computing task scheduling, focusing on optimizing both makespan and energy usage. They introduced the Genetic Algorithm and Energy-Conscious Scheduling Heuristic (GAECS) algorithm to tackle job scheduling in cloud environments. Combining GA and ECSH aims to reduce makespan, improve resource allocation efficiency, and enhance overall system performance. However, achieving optimal results requires careful tuning of the GA and ECSH algorithm parameters, which can be challenging for users unfamiliar with optimization techniques.

5.3 Hyper-heuristic approach

Hyper-heuristics represent an advanced automated search methodology designed to tackle the complexity of scheduling tasks in cloud environments. Instead of directly navigating the search space, hyper-heuristics operate within a search space of low-level heuristics, enabling a more flexible and adaptive problem-solving approach. This technique addresses computational search problems by selecting, combining, generating, or adapting simpler heuristics, often leveraging machine learning techniques [106]. Unlike MHs, which primarily focus on searching the solution space, hyper-heuristics focus on exploring diverse heuristics, enabling them to adapt to various problem domains [107]. The scheduling process becomes more flexible and robust by using hyper-heuristics as they can dynamically adjust their strategy to optimize performance in complex and variable environments [108]. The hyper-heuristic is classified as [108] follows:

(a) Heuristic search space is classified into two parts:
- Heuristic selection: Methods for choosing or selecting existing heuristics.
- Heuristic generation: Methods for creating new heuristics from existing components.

(b) The feedback source during learning is classified into two parts:
- Online learning: Learning occurs as the algorithm solves a specific problem instance.
- Offline learning: Acquiring knowledge in the form of rules or programs from a set of training instances, to generalize to solve unseen instances.

Hyper-heuristics are able to handle diverse workloads, dynamic nature, and heterogeneous resources in cloud computing environments. By leveraging a combination of selection and heuristic construction, hyper-heuristics can effectively adapt to the diverse nature of tasks and system configurations [109].

Hyper heuristics, which integrate online and offline learning, enhance their versatility by refining strategies in real-time and storing valuable insights for future use. By incorporating RIL, these frameworks associate specific heuristics with successful outcomes, improving performance across various optimization problems such as resource allocation and load balancing [108]. Future research aims to boost efficiency, scalability, and knowledge transfer between task domains, making hyper-heuristics a key component in automating decision-making processes in complex environments [109].

5.3.1 Review of hyper-heuristic approach

This section reviews articles on resource allocation and task scheduling algorithms that are based on hyper-heuristic approach; Table 4 lists the algorithms used in cloud computing environments.

Yin et al. [110] proposed a hyper-heuristic reinforcement learning (HHRL)-based approach to enhance task sequence completion time for complicated and dynamic cloud service scheduling tasks. This method integrates population diversity and maximum time to determine job scheduling and low-level heuristic strategies. It also introduces a linear regression-based approach to estimate task complexity, tracking the completion of 100 tasks in each category and examining their linear relationships. Nonetheless, the study does not address the potential limitations or constraints in the reward table setting phase of HHRL, particularly in the selection of low-level heuristic strategies and the integration of maximum time and population diversity.

Table 4: HHAs used in resource allocating and task scheduling

Algorithm	Simulation/tool	Objective(s)/key performance metrics	Quantitative results	Sample size (no. of cloud nodes, tasks, or datasets)	Main focus area	Limitations
HHRL-based approach [110]	CloudSim	Task sequence completion time	Improves task sequence completion time by tracking the completion of 100 tasks to estimate difficulty	100 tasks per category	Task scheduling	Potential limitations of reward table setting stage not discussed
Honey Bee Algorithm [111]	CloudSim	Makespan time, load distribution, and degree of imbalance	Ensures balanced load distribution among VMs, minimize makespan time	Not specified	Task scheduling	Limited comparison metrics, potential areas for enhancement not identified
Genetic programming hyper-heuristics with human-designed rules (GPHH) [112]	Java version 8	Energy consumption	Outperforms existing methods of resource allocation and minimizes energy consumption in container-based clouds	Not specified	Resource allocation	Focus solely on power consumption, ignoring other performance indicators
Multi-objective artificial bee colony with Q-learning (MOABCQ) [8]	CloudSim	Cost, makespan, and resource utilization	Minimize makaspan, reduce cost, maximize resource utilization, and improve VM throughput	Google Cloud Jobs (GoCJ), and Synthetic workload	Resource allocation, and task scheduling	The MOABCQ method may not always be optimal, and its performance may vary across different datasets
Queue-priority algorithm with PSO [113]	Java programming language	Scalability, and operational costs	Enhances task scheduling, enhances scalability, and minimize operational costs	The workloads tested included 1,000, 10,000, 100,000, 500,000, and higher workloads up to 2,500,000 inbound messages	Task scheduling	Lack of comprehensive evaluation against other industry methods
HSQ-StudGA (Q-learning with genetic algorithms) [114]	MATLAB	Solution quality	Integrates Q-learning with genetic algorithms to improve solution quality and search capabilities	Not specified	Task scheduling	Potential challenges of implementing Q-learning not addressed
PDCS with hyper-heuristic [115]	Java Eclipse IDE and a JADE	MCT	Reduce MCT by efficiently adapting scheduling rules for dynamic systems	The sample size includes the evaluation of the algorithm's performance on benchmark functions, such as f01 to f14	Task scheduling	The rapidly changing environment poses challenges in implementing conventional predictive methods

Gupta et al. [111] proposed an algorithm to achieve balanced load distribution across VMs, aiming to reduce the time of makespan. The algorithm offers equitable scheduling frameworks by employing a honey bee load balancing and improvement detection operator to determine the appropriate low-level heuristic for enhanced candidate solutions. Experimental results demonstrate its efficiency compared to existing heuristic-based scheduling procedures. However, further analysis and evaluation are needed to identify potential limitations or areas for enhancement. Additionally, the comparison of the proposed algorithm was limited to specific metrics and did not address the full spectrum of cloud task scheduling optimization metrics.

A previous research [112] presented a new two-level container allocation issue in cloud computing and suggested a hybrid method utilizing genetic programming hyper-heuristics and human-designed rules to reduce energy usage notably. The proposed method surpasses current rules by considering various characteristics to choose and generate VMs, resulting in more effective resource allocation in container-based clouds, thereby reducing energy consumption. However, this study focuses on reducing power consumption as the primary metric while ignoring other important performance indicators in container-based cloud environments.

Kruekaew and Kimpan [8] developed a multi-objective approach known as MOABCQ, which uses the artificial bee colony algorithm, the Q-learning algorithm, and RIL technique, to enhance scheduling and resource utilization, maximize VM throughput, and create load balancing between VMs based on makespan, cost, and resource utilization. The MOABCQ approach demonstrated strong search capabilities and quick convergence. However, the MOABCQ method may not always be optimal and its performance may vary across different datasets.

Freire et al. [113] proposed a queue-priority algorithm that utilizes a new heuristic and PSO to optimize task scheduling in integration platforms, particularly for managing large data volumes in integration processes. The algorithm enhances the scalability of cloud computational resources and reduces business operational costs. The experimental results confirmed the efficacy of the proposed algorithm for improving the execution of integration processes with high data volumes. However, the study does not offer an exhaustive evaluation of the suggested algorithm's performance about alternative methods in the sector, nor does it compare it with other task scheduling methods that are currently in use.

Tong et al. [114] proposed a novel intelligent HHA called HSQ-StudGA, which combines machine learning technology, specifically Q-learning, with GAs to solve optimization problems. It aims to enhance the performance and quality of arithmetic solutions for such problems. The HHA improves the search capability of the original algorithm solution and effectively enhances the overall quality of the solution. However, the study does not address the potential challenges or limitations of implementing the Q-learning method in the algorithm.

The mean completion time (MCT) was found to be minimized more effectively by Bouazza et al. [115] when they proposed a Product-Driven Control System (PDCS) based on smart products for the dynamic scheduling of production systems. The approach integrates a hyper-heuristic with a generic decisional strategy model, enabling efficient switching between scheduling rules and enhancing global performance and system reactivity. The proposed approach shows superior performance in minimizing mean completion time. However, the rapidly changing environment poses challenges in implementing conventional predictive methods. Table 4 shows the hyper-heuristic algorithms used in resource allocating and scheduling.

6 Discussion

Resource allocation involves distributing resources to cloud applications over the internet, while task scheduling organizes tasks to ensure efficient resource utilization. Both processes significantly impact the performance and quality of cloud applications. This section explores widely adopted strategies and algorithms for resources allocation and task scheduling in the cloud environments. We categorize existing approaches into mathematical, heuristic, and hyper-heuristic methodologies, each playing a crucial role in improving cloud performance and reducing costs. Through a systematic review of 100 studies from leading scientific databases, we provide valuable insights into these techniques. Our analysis of literature from 2019 to 2023 identifies key trends, challenges, and opportunities in this rapidly evolving field.

Mathematical approaches, while providing optimal solutions, demonstrate limited practical applicability in large-scale cloud environments. The literature shows that methods such as linear programming and game theory models offer precise solutions but often struggle with scalability issues. For instance, the Stackelberg game model showed impressive results in revenue maximization but was tested only in simplified scenarios with limited service providers and VM instances [104].

Heuristic approaches, particularly MH algorithms, emerge as the most widely adopted solutions in current cloud environments. Notable algorithms such as PSO, GA, and ACO demonstrate superior adaptability and near-optimal solutions. The trend toward hybrid approaches is particularly noteworthy, with combinations such as OWPSO and HCSOA-TS showing improved performance across multiple metrics including makespan, cost, and resource utilization [105]. Hyper-heuristics can effectively manage the complexity of scheduling tasks in a cloud environment by leveraging diverse heuristics to explore different parts of the solution space [116]. By dynamically selecting and tuning low-level heuristics, hyper-heuristics can optimize the allocation of cloud resources, leading to better utilization and reduced operational costs [117]. Moreover, the highly dynamic nature of the cloud environment can be effectively managed using hyper-heuristics. This is due to the latter's potential to adapt to real-time changes ensuring efficient scheduling under varying conditions [118]. Studies implementing HHRL and MOABCQ demonstrate promising results in handling dynamic environments, though their implementation complexity remains a concern.

Deep RIL-based approaches [70] demonstrated strong performance for container-based clouds, showing improved resource utilization and execution time metrics. Similarly, the deep RIL approach [78] effectively optimized multiple QoS parameters including energy consumption, time, and SLA violations. The hybrid MH solutions such as FPSO-GA approach [39], which combines GA and fuzzy PSO, showed superior performance in load balancing and energy efficiency. The AMO-TLBO algorithm [100] incorporated adaptive teaching factors and self-motivated learning, proving especially effective at increasing utilization while reducing space and cost.

Hybrid approaches, such as the combination of neural networks and classification techniques described in [68], integrate neural network-based classification with genetic algorithm optimization to achieve notable performance gains. Similarly, multi-objective optimization methods like RAA-PI-NSGAII [56] have shown strong results in improving resource utilization, reducing computation time, aligning resource capabilities, and enhancing the overall quality of solution sets.

It is worth noting that recent research reveals a clear shift toward multi-objective optimization, with effective strategies integrating multiple approaches rather than relying on a single method. These strategies succeed by adapting to changing workload conditions while balancing objectives such as resource utilization, energy efficiency, response time, QoS parameters, and load balancing requirements. While traditional metrics such as makespan and cost remain important, energy efficiency and other advanced considerations are increasingly prioritized [17]. A recurring theme across studies is the challenge of scalability. While many algorithms perform well in controlled environments, their effectiveness often diminishes with increasing system size and complexity. This highlights the need for more robust solutions capable of handling real-world cloud environments [119].

6.1 Future trends in resource allocation and task scheduling

As cloud computing continues to grow, the following emergent trends in resource allocation and task scheduling are recognized from the reviewed literature [119].

6.1.1 Hybrid and hyper-heuristic techniques for dynamic environments

Hybrid methods that integrate various algorithms or heuristics are increasingly popular for achieving near-optimal solutions under diverse scenarios. Trends include [114,116]:

- Hyper-heuristics: These techniques dynamically select and tune low-level heuristics to adapt to changing conditions, ensuring efficient resource allocation. Their potential to handle real-time changes in task scheduling offers a promising direction for future cloud applications.
- Hybrid MH: Combinations such as OWPSO and HCSOA-TS demonstrate enhanced performance across various metrics, suggesting that hybridization will remain a cornerstone of future algorithm development.

6.1.2 Multi-objective optimization

Future research will likely prioritize simultaneous achievement of multiple objectives. While traditional metrics such as cost and makespan remain essential, there is growing interest in incorporating the following:
- Energy efficiency: With sustainability becoming a priority, minimizing energy consumption will be critical. Approaches such as the hybrid genetic programming method [112] emphasize energy usage reduction, setting the stage for more energy-conscious scheduling algorithms.
- QoS considerations: Metrics such as latency, reliability, and availability are gaining prominence, especially in real-time applications.
- Load balancing and resource utilization: Ensuring optimal use of resources without overloading VMs or cloud instances will remain a core focus.

6.1.3 Integrating AI and machine learning

AI and ML are expected to play a transformative role in cloud computing:
- RIL approaches: Techniques such as HHRL [110] and MOABCQ [8] show potential for adapting to dynamic cloud environments, though simplifying their implementation complexities will be a future challenge.
- Predictive analytics: Employing linear regression or other predictive models to estimate task difficulty or resource needs can further enhance scheduling accuracy.

6.1.4 Emergence of edge-cloud collaboration

As edge computing expands, resource allocation and task scheduling need to tackle distributed and hierarchical environments.
- Algorithms must evenly distribute computation loads between cloud data centers and edge devices.
- Addressing latency-sensitive applications such as IoT and real-time analytics will drive innovations in hybrid edge-cloud scheduling models.

6.1.5 Focus on green and sustainable cloud computing

As energy consumption becomes a global concern, research will increasingly prioritize sustainable solutions. This includes
- Developing algorithms that optimize resource usage without compromising performance.
- Integrating renewable energy sources into cloud infrastructure and aligning scheduling tasks with energy availability patterns.

6.1.6 Scalability and real-world applicability

Scalability remains a critical challenge for most algorithms, especially in large-scale, heterogeneous cloud environments. Future efforts must focus on

- Scalable algorithms: Solutions capable of maintaining efficiency across increasing system sizes and complexity.
- Benchmarking and real-world testing: Moving beyond controlled simulations to test algorithms in diverse and realistic cloud scenarios.

Future research should prioritize developing hybrid models that combine various scheduling and allocation strategies for a comprehensive approach to these challenges. Fostering collaboration between academia and industry can accelerate the advancement of best practices and innovative solutions. Moreover, utilizing machine learning and predictive analytics can offer insights into usage patterns, enabling proactive resource management that anticipates demand [119].

7 Study characteristics

Among the 100 articles analyzed in this study, 55% primarily focused on task scheduling algorithms, 35% on resource allocation techniques, and 10% addressed both. Most of the studies (75%) evaluated the proposed algorithms using simulation-based techniques, especially CloudSim. Ten percent of the studies were predominantly theoretical or analytical, while 15% included experiments conducted in real cloud settings. MH emerged as the most frequently employed algorithmic approach in 45% of the investigations. Twenty percent of the articles used mathematical optimization techniques, 25% used heuristics, and 10% used hyper-heuristics. Thirty percent of the reviewed studies focused on multi-objective optimization. While makespan was nearly universally reported, fewer studies evaluated metrics such as reliability, security, or user satisfaction that may be equally important in practice. Figure 6 shows the number of articles that are modeling the task scheduling, resource allocation, or both in the cloud computing environment. Figure 7 illustrates the number of metrics iterations considered in previous articles.

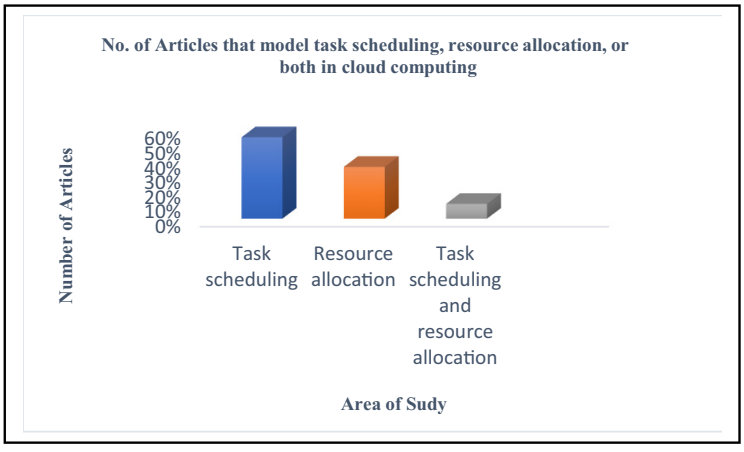

Figure 6: Illustrates articles that model task scheduling, resource allocation, or both in cloud computing (created by the authors).

As shown in Figure 7, some algorithms focus extensively on makespan, energy efficiency, and maximizing resource utilization. Various scheduling techniques prioritize factors such as reducing cost, response time, and increasing throughput within cloud environments. Additionally, numerous algorithms emphasize latency, reliability, and load balancing. Furthermore, instead of using real-time datasets, most authors employed generated datasets. Additionally, it is noted that, when they composed their assessment, CloudSim was the simulator utilized the most.

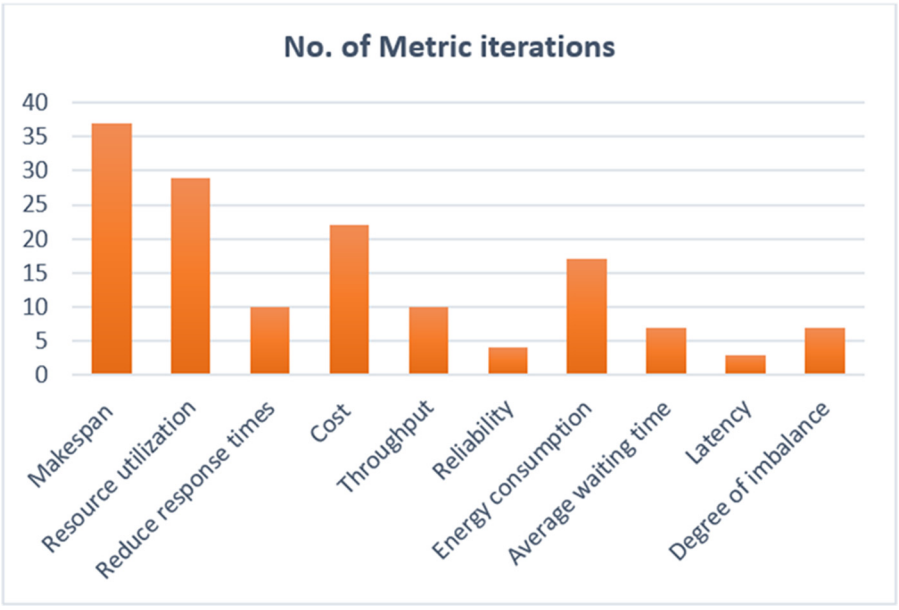

Figure 7: Illustrates the number of metrics iterations considered in previous algorithms (created by the authors).

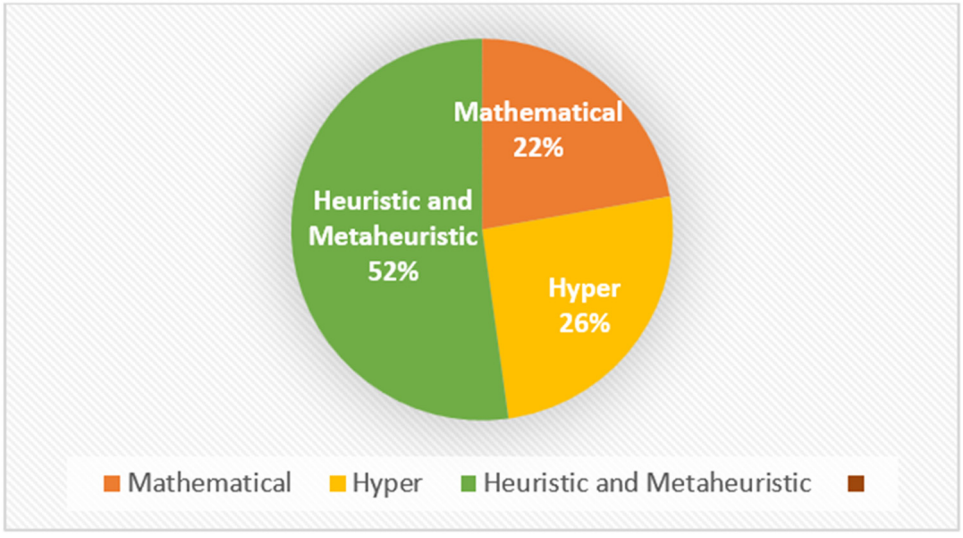

Figure 8: Percentage of resource allocating and scheduling algorithms from 2019 to 2023 according to our systematic review (created by the authors).

Tables 2–4 in Section 4, despite many studies in the field of allocation and scheduling in cloud computing, many existing methods cannot effectively balance multiple and conflicting objectives simultaneously. Furthermore, traditional optimization algorithms are usually static and lack of adaptability to dynamic changes in the cloud environment. Additionally, traditional algorithms often assume that resources and tasks are homogeneous, which does not reflect the practical reality where resources are diverse and demands are variable. Figure 8 shows the percentage of resource allocating and scheduling algorithms from 2019 to 2023 according to our systematic review. While Figure 9 shows the percentage of language/simulator usage according to our systematic review.

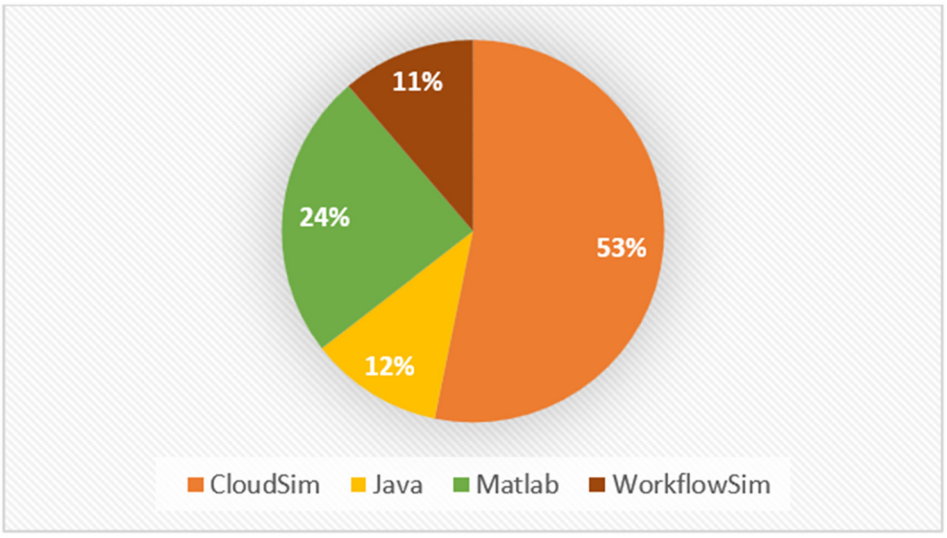

Figure 9: Percentage of language/simulator usage according to our systematic review (created by the authors).

8 Challenges and research gaps

Despite significant research, several challenges persist in previous literature [111,120].

1. Scalability: Developing algorithms that can efficiently handle the increasing scale of cloud infrastructure and workloads.
2. Adaptability: Developing solutions that swiftly adjust to the dynamic nature of cloud environments through increased hybridization and the integration of optimization algorithms with technologies such as machine learning.
3. Multi-objective optimization: Balancing multiple, often conflicting, objectives simultaneously.
4. Integration with emerging technologies: Incorporating new paradigms such as edge computing, serverless computing, and AI into resource management strategies.
5. Security and privacy: Ensuring robust security measures and data privacy in resource allocation frameworks is crucial, especially as data breaches become increasingly sophisticated. Developing algorithms that protect sensitive information while maintaining efficiency poses a significant challenge.
6. Cost efficiency: Optimizing resource utilization to minimize operational costs while meeting performance requirements is a pressing concern for cloud service providers. This necessitates the creation of cost-effective scheduling algorithms that can assess economic factors alongside performance metrics.
7. Real-time decision making: As workloads fluctuate, the ability to make resource allocation decisions in real-time becomes essential. Research into low-latency decision-making frameworks that can respond to immediate demands without compromising service quality is needed to enhance user experience.
8. User-centric resource management: Tailoring resource management strategies to meet the specific needs of diverse user groups remains a challenge. Developing customizable frameworks allowing users to dictate their resource preferences can improve satisfaction and productivity.

9 Conclusion

This SLR contributes to the field of cloud computing by providing a comprehensive analysis of recent developments in resource allocation and task scheduling algorithms. Our theoretical analysis reveals the evolution from traditional mathematical approaches to more sophisticated hybrid solutions that combine multiple methodologies. The classification framework we developed helps bridge the gap between theoretical

foundations and practical implementations by organizing approaches into mathematical, hyper, and heuristic categories. The research makes several key contributions to the field. First, it systematically classifies resource management approaches into mathematical, heuristic, and hyper-heuristic approaches, enabling researchers to better understand the relationships between different methodologies. Second, it identifies makespan, cost, energy, and resource utilization as the primary optimization parameters in current task scheduling research. Third, it reveals a significant reliance on synthetic datasets and CloudSim simulations, highlighting the need for more real-world implementation studies. From a practical perspective, our SLR offers valuable insights for cloud service providers and researchers. The review demonstrates that integrating heuristic and hyper-heuristic approaches can lead to near-optimal solutions with enhanced scalability and adaptability, making them particularly well-suited for large-scale, dynamic cloud environments. Additionally, the insights into multi-objective optimization highlight opportunities to balance competing priorities such as cost efficiency, energy consumption, and resource utilization, paving the way for more efficient and sustainable cloud operations. The identification of CloudSim as the predominant simulation tool also helps practitioners choose appropriate testing environments for new implementations. However, several limitations must be acknowledged. The heavy reliance on synthetic datasets rather than real-time data limits the direct applicability of many findings to production environments. Additionally, while CloudSim provides a standardized testing environment, its widespread use may not fully capture the complexity of real-world cloud systems. The focus on specific performance metrics may also overlook other important factors in cloud resource management. Looking forward, researchers and practitioners can build upon these findings to develop more efficient and effective resource management strategies that address the challenges of dynamic and heterogeneous cloud environments while balancing performance optimization with cost considerations. Future work should focus on validating theoretical approaches with real-world implementations and expanding the range of performance metrics considered in resource management strategies.

Acknowledgements: The authors gratefully acknowledge the financial support of the Laboratory of the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia and the University of Anbar, Iraq.

Funding information: Fundamental Research Grant Scheme (FRGS/1/2023/ICT07/UKM/02/3) under Ministry of Higher Education Malaysia.

Authors contributions: Waleed Kareem drafted the original manuscript, conceptualized it, and performed literature analysis. Dr Akram and Dr Mohd Zakree: conceptualization and methodology, scientific advice, supervision, drafting, review, and editing. Dr. Esam Taha: scientific advice and supervision. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors have no conflicts of interest to declare that are relevant to the content of this article.

Institutional review board statement: Not applicable.

Data availability statement: Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

[1] Murad SA, Muzahid AJM, Azmi ZRM, Hoque MI, Kowsher M. A review on job scheduling technique in cloud computing and priority rule based intelligent framework. King Saud bin Abdulaziz Univ. 2022;34(6):2309–31. doi: 10.1016/j.jksuci.2022.03.027.

[2] Al-Jumaili AHA, Muniyandi RC, Hasan MK, Singh MJ, Paw JKS. Intelligent transmission line fault diagnosis using the Apriori associated rule algorithm under cloud computing environment. J Auton Intell. 2023;6(1):640. doi: 10.32629/jai.v6i1.640.
[3] Liu X, Buyya R. Resource management and scheduling in distributed stream processing systems: a taxonomy, review, and future directions. Assoc Comput Machinery. 2020;53(3):50. doi: 10.1145/3355399.
[4] AL-Jumaili AHA, Muniyandi RC, Hasan MK, Singh MJ, Paw JKS, Amir M. Advancements in intelligent cloud computing for power optimization and battery management in hybrid renewable energy systems: A comprehensive review. Energy Rep. 2023;10:2206–27. doi: 10.1016/j.egyr.2023.09.029.
[5] Konjaang JK, Xu L. Meta-heuristic approaches for effective scheduling in infrastructure as a service cloud: a systematic review. J Netw Syst Manag. Apr. 2021;29:15. doi: 10.1007/s10922-020-09577-2.
[6] AL-Jumaili AHA, Mashhadany YIA, Sulaiman R, Alyasseri ZAA. A conceptual and systematics for intelligent power management system-based cloud computing: Prospects, and challenges. Appl Sci (Switz). Nov. 2021;11(21):9820. doi: 10.3390/app11219820.
[7] Fawad A, Saad Zahoor M, Ellahi E, Yerasuri S, Muniandi B, Balasubramanian S. Efficient workload allocation and scheduling strategies for AI-intensive tasks in cloud infrastructures. Nanjing, China: Power system technology, State grid electric power research institute; 2023. doi: 10.52783/pst.160.
[8] Kruekaew B, Kimpan W. Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning. IEEE Access. 2022;10:17803–18. doi: 10.1109/ACCESS.2022.3149955.
[9] AL-Gburi AFJ, Nazri MZA, Bin Yaakub MR, Alyasseri ZAA. A systematic review of symbiotic organisms search algorithm for data clustering and predictive analysis. J Intell Syst. 2024;33(1):20230267. doi: 10.1515/jisys-2023-0267.
[10] Karimi-Mamaghan M, Mohammadi M, Meyer P, Karimi-Mamaghan AM, Talbi EG. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. Eur J Oper Res. 2022;296(2):393–422. doi: 10.1016/j.ejor.2021.04.032.
[11] Panwar SS, Rauthan MMS, Barthwal V. A systematic review on effective energy utilization management strategies in cloud data centers. J Cloud Comput. 2022;11(1):95. doi: 10.1186/s13677-022-00368-5.
[12] Ben Alla H, Ben Alla S, Ezzati A, Touhafi A. A novel multiclass priority algorithm for task scheduling in cloud computing. J Supercomputing. Oct. 2021;77(10):11514–55. doi: 10.1007/s11227-021-03741-4.
[13] Priya V, Sathiya Kumar C, Kannan R. Resource scheduling algorithm with load balancing for cloud service provisioning. Appl Soft Comput J. Mar. 2019;76:416–24. doi: 10.1016/j.asoc.2018.12.021.
[14] Del Gallo M, Mazzuto G, Ciarapica FE, Bevilacqua M. Artificial intelligence to solve production scheduling problems in real industrial settings: systematic literature review. Electronics. 2023;12(23):4732. doi: 10.3390/electronics12234732.
[15] Fadhil HM. Optimizing task scheduling and resource allocation in computing environments using metaheuristic methods. Fusion: Pract Appl. 2024;15(1):157–79. doi: 10.54216/FPA.150113.
[16] Jia R, Yang Y, Grundy J, Keung J, Hao L. A systematic review of scheduling approaches on multi-tenancy cloud platforms. Inf Software Technol. 2021 Apr;132:106478. https://doi.org/10.1016/j.infsof.2020.106478.
[17] Negi S, Singh DP, Rauthan MMS. A systematic literature review on soft computing techniques in cloud load balancing network. Springer; 202315:800–38. doi: 10.1007/s13198-023-02217-3.
[18] Houssein EH, Gad AG, Wazery YM, Suganthan PN. Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. Swarm Evol Comput. Apr. 2021;62:100841. doi: 10.1016/j.swevo.2021.100841.
[19] Zhou J, Lilhore UK, Hai T, Simaiya S, Jawawi DN, Alsekait D, et al. Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing. J Cloud Comput. Dec. 2023;12:85. doi: 10.1186/s13677-023-00453-3.
[20] Arunarani AR, Manjula D, Sugumaran V. Task scheduling techniques in cloud computing: A literature survey. Future Gener Comput Syst. Feb. 2019;91:407–15. doi: 10.1016/j.future.2018.09.014.
[21] Chhabra M, Basheer S. Recent task scheduling-based heuristic and meta-heuristics methods in cloud computing: a review. In Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022, Institute of Electrical and Electronics Engineers Inc; 2022. p. 2236–42. doi: 10.1109/IC3I56241.2022.10073445.
[22] Saif MAN, Niranjan SK, Al-ariki HDE. Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis. Wirel Netw. May 2021;27(4):2829–66. doi: 10.1007/s11276-021-02614-1.
[23] Kamatar M, Madhavi B. A comparative study on resource aware allocation and load balancing techniques for cloud computing. Grenze Int J Eng Technol. 2023;9(1):1–5.
[24] Belgacem A. Dynamic resource allocation in cloud computing: analysis and taxonomies. Computing. Mar. 2022;104(3):681–710. doi: 10.1007/s00607-021-01045-2.
[25] Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ. 2021;372:n71. doi: 10.1136/bmj.n71.
[26] Zhao Y, Pinto Llorente AM, Sánchez Gómez MC. Digital competence in higher education research: A systematic literature review. Comput Educ. Jul. 2021;168:104212. doi: 10.1016/j.compedu.2021.104212.
[27] Shiekh S, Shahid M, Sambare M, Haidri RA, Yadav DK. A load-balanced hybrid heuristic for allocation of batch of tasks in cloud computing environment. Int J Pervasive Comput Commun. 2022;19(5):756–81. doi: 10.1108/IJPCC-06-2022-0220.
[28] Shirvani MH, Talouki RN. A novel hybrid heuristic-based list scheduling algorithm in heterogeneous cloud computing environment for makespan optimization. Parallel Comput. Dec. 2021;108:102828. doi: 10.1016/j.parco.2021.102828.

[29] Chhabra A, Huang KC, Bacanin N, Rashid TA. Optimizing bag-of-tasks scheduling on cloud data centers using hybrid swarm-intelligence meta-heuristic. J Supercomputing. May 2022;78(7):9121–83. doi: 10.1007/s11227-021-04199-0.
[30] Alyas T, Ghazal TM, Alfurhood BS, Issa GF, Thawabeh OA, Abbas Q. Optimizing resource allocation framework for multi-cloud environment. Comput Mater Continua. 2023;75(2):4119–36. doi: 10.32604/cmc.2023.033916.
[31] Mangalampalli S, Karri GR, Kose U. Multi objective trust aware task scheduling algorithm in cloud computing using whale optimization. J King Saud Univ - Comput Inf Sci. Feb. 2023;35(2):791–809. doi: 10.1016/j.jksuci.2023.01.016.
[32] Tanha M, Hosseini Shirvani M, Rahmani AM. A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments. Neural Comput Appl. Dec. 2021;33(24):16951–84. doi: 10.1007/s00521-021-06289-9.
[33] Kumar M, Sharma SC, Goel A, Singh SP. A comprehensive survey for scheduling techniques in cloud computing. J Netw Comput Appl. 2019;143:1–33. doi: 10.1016/j.jnca.2019.06.006.
[34] Rachna MS, Namrata MS, Diksha MS. Resource allocation in cloud. Int J Res Appl Sci Eng Technol. Feb. 2022;10(2):1395–9. doi: 10.22214/ijraset.2022.40517.
[35] Mahdi ET, Awad WK, Rasheed MM, Mahdi AT. Proposed security system for cities based on animal recognition using IoT and clouds. In 2023 16th International Conference on Developments in eSystems Engineering (DeSE). IEEE; 2023. p. 834–9.
[36] Kumar M, Sharma SC. PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing. Neural Comput Appl. Aug. 2020;32(16):12103–26. doi: 10.1007/s00521-019-04266-x.
[37] Amer DA, Attiya G, Zeidan I, Nasr AA. Elite learning Harris Hawks optimizer for multi-objective task scheduling in cloud computing. J Supercomputing. Feb. 2022;78(2):2793–818. doi: 10.1007/s11227-021-03977-0.
[38] Sihwail R, Omar K, Ariffin KAZ, Tubishat M. Improved Harris Hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. IEEE Access. 2020;8:121127–45. doi: 10.1109/ACCESS.2020.3006473.
[39] Mirmohseni SM, Tang C, Javadpour A. FPSO-GA: A fuzzy metaheuristic load balancing algorithm to reduce energy consumption in cloud networks. Wirel Pers Commun. Dec. 2022;127(4):2799–821. doi: 10.1007/s11277-022-09897-3.
[40] Saidi K, Bardou D. Task scheduling and VM placement to resource allocation in cloud computing: challenges and opportunities. Clust Comput. 2023;26(5):3069–87. doi: 10.1007/s10586-023-04098-4.
[41] Golec M, Hatay ES, Golec M, Uyar M, Golec M, Gill SS. Quantum cloud computing: Trends and challenges. J Econ Technol. Nov. 2024;2:190–9. doi: 10.1016/j.ject.2024.05.001.
[42] AL-Jumaili AHA, Muniyandi RC, Hasan MK, Paw JKS, Singh MJ. Big data analytics using cloud computing based frameworks for power management systems: status, constraints, and future recommendations. Sensors. 2023;23(6):2952. doi: 10.3390/s23062952.
[43] Al-Mahruqi AAH, Morison G, Stewart BG, Athinarayanan V. Hybrid heuristic algorithm for better energy optimization and resource utilization in cloud computing. Wirel Pers Commun. May 2021;118(1):43–73. doi: 10.1007/s11277-020-08001-x.
[44] Bu T, Huang Z, Zhang K, Wang Y, Song H, Zhou J, et al. Task scheduling in the internet of things: challenges, solutions, and future trends. Clust Comput. Feb. 2024;27(1):1017–46. doi: 10.1007/s10586-023-03991-2.
[45] Ziyath SPM, Senthilkumar S. MHO: meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services. J Ambient Intell Human Comput. 2021;12:6629–38. doi: 10.1007/s12652-020-02282-7.
[46] Jena UK, Das PK, Kabat MR. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. J King Saud Univ - Comput Inf Sci. Jun. 2022;34(6):2332–42. doi: 10.1016/j.jksuci.2020.01.012.
[47] Awad WK, Mahdi ET. Tasks scheduling techniques in cloud computing. In 2022 3rd Information technology to enhance e-learning and other application (IT-ELA); 2022. p. 94–98). IEEE. doi: 10.1109/IT-ELA57378.2022.10107956.
[48] Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A. A survey on new generation metaheuristic algorithms. Comput Ind Eng. Nov. 2019;137:106040. doi: 10.1016/j.cie.2019.106040.
[49] Krishnasamy KG, Periasamy K, Veerappan PM, Thangavel G, Lamba R, Muthusamy S. A pair-task heuristic for scheduling tasks in heterogeneous multi-cloud environment. Computers & Industrial Engineering; 2022. doi: 10.21203/rs.3.rs-1903846/v1.
[50] Yassen ET, Ayob M, Jihad AA, Nazri MZA. A self-adaptation algorithm for quay crane scheduling at a container terminal. IAES Int J Artif Intell. Dec. 2021;10(4):919–29. doi: 10.11591/IJAI.V10.I4.PP919-929.
[51] Zhu Z, Peng J, Liu K, Zhang X. A game-based resource pricing and allocation mechanism for profit maximization in cloud computing. Soft Comput. Mar. 2020;24(6):4191–203. doi: 10.1007/s00500-019-04183-0.
[52] Hamzeh H, Meacham S, Khan K, Phalp K, Stefanidis A. MRFS: A multi-resource fair scheduling algorithm in heterogeneous cloud computing. In Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020. Institute of Electrical and Electronics Engineers Inc.; Jul. 2020. p. 1653–60. doi: 10.1109/COMPSAC48688.2020.00-18.
[53] Scavuzzo L, Aardal K, Lodi A, Yorke-Smith N. Machine learning augmented branch and bound for mixed integer linear programming. Math Program. 2024;1653–60. doi: 10.1007/s10107-024-02130-y.
[54] Clautiaux F, Ljubić I. Last fifty years of integer linear programming: A focus on recent practical advances. Eur J Oper Res. 2024. doi: 10.1016/j.ejor.2024.11.018.
[55] Jaber A, Younes R, Lafon P, Khoder J. A review on multi-objective mixed-integer non-linear optimization programming methods. Eng. Aug. 2024;5(3):1961–79. doi: 10.3390/eng5030104.
[56] Chen J, Du T, Xiao G. A multi-objective optimization for resource allocation of emergent demands in cloud computing. J Cloud Comput. Dec. 2021;10(1):1961–79. doi: 10.1186/s13677-021-00237-7.

[57] Swatthong N, Aswakul C. Optimal cloud orchestration model of containerized task scheduling strategy using integer linear programming: Case studies of iotcloudserve@tein project. Energ (Basel). Aug. 2021;14(15):4536. doi: 10.3390/en14154536.
[58] Yadav M, Mishra A. An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment. J Cloud Comput. Dec. 2023;12:8. doi: 10.1186/s13677-023-00392-z.
[59] Tai KY, Lin FYS, Hsiao CH. An integrated optimization-based algorithm for energy efficiency and resource allocation in heterogeneous cloud computing centers. IEEE Access. 2023;11:53418–28. doi: 10.1109/ACCESS.2023.3280930.
[60] Al-Asaly MS, Bencherif MA, Alsanad A, Hassan MM. A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment. Neural Comput Appl. Jul. 2022;34(13):10211–28. doi: 10.1007/s00521-021-06665-5.
[61] Ghobaei-Arani M, Souri A. LP-WSC: a linear programming approach for web service composition in geographically distributed cloud environments. J Supercomputing. May 2019;75(5):2603–28. doi: 10.1007/s11227-018-2656-3.
[62] Rawat PS, Dimri P, Kanrar S, Saroha GP. Optimize task allocation in cloud environment based on Big-Bang Big-Crunch. Wirel Pers Commun. Nov. 2020;115(2):1711–54. doi: 10.1007/s11277-020-07651-1.
[63] Shi W, Tang D, Zou P. Research on cloud enterprise resource integration and scheduling technology based on mixed set programming. Tehnicki Vjesn. Nov. 2021;28(6):2027–35. doi: 10.17559/TV-20210718091658.
[64] Almojel NA, Ahmed AES. Tasks and resources allocation approach with priority constraints in cloud computing. Int J Grid High Perform Comput (IJGHPC). 2022;14(1):1–17. doi: 10.4018/ijghpc.301584.
[65] Brandwajn A, Begin T. First-come-first-served queues with multiple servers and customer classes. Perform Evaluation. Apr. 2019;130:51–63. doi: 10.1016/j.peva.2018.11.001.
[66] Saudi Computer Society. Institute of electrical and electronics engineers. Saudi Arabia Section, Institute of Electrical and Electronics Engineers. Region 8, and Institute of Electrical and Electronics Engineers. 2nd International Conference on Computer Applications & Information Security (ICCAIS' 2019). Riyadh, Kingdom of Saudi Arabia: May, 2019. doi: 10.1109/CAIS.2019.8769534.
[67] Alsadie D. Virtual machine placement methods using metaheuristic algorithms in a cloud environment-a comprehensive review. IJCSNS Int J Comput Sci Netw Secur 22(4):147–58. doi: 10.22937/IJCSNS.2022.22.4.19.
[68] Manavi M, Zhang Y, Chen G. Resource allocation in cloud computing using genetic algorithm and neural network. In 2023 IEEE 8th International Conference on Smart Cloud (SmartCloud); 2023. p. 25–32. http://arxiv.org/abs/2308.11782.
[69] Ghazy N, Abdelkader A, Zaki MS, Eldahshan KA. An ameliorated Round Robin algorithm in the cloud computing for task scheduling. Bull Electr Eng Inform. Apr. 2023;12(2):1103–14. doi: 10.11591/eei.v12i2.4524.
[70] Zhu L, Wu F, Hu Y, Huang K, Tian X. A heuristic multi-objective task scheduling framework for container-based clouds via actor-critic reinforcement learning. Neural Comput Appl. May 2023;35(13):9687–710. doi: 10.1007/s00521-023-08208-6.
[71] Jihad AA, Faraj Al-Janabi ST, Yassen ET. A survey on provisioning and scheduling algorithms for scientific workflows in cloud computing. In AIP Conference Proceedings. American Institute of Physics Inc; Oct. 2022, doi: 10.1063/5.0112122.
[72] Awad WK, Mahdi ET, Rashid MN. Features extraction of fingerprints based bat algorithms. Int J Tech Phys Probl Eng. 2022;14(4):274–9.
[73] Shami TM, Grace D, Burr A, Mitchell PD. Single candidate optimizer: a novel optimization algorithm. Evol Intell. 2024;17:863–87. doi: 10.1007/s12065-022-00762-7.
[74] Raidl GR, Puchinger J, Blum C, Raidl GR, Puchinger J, Blum C. Metaheuristic hybrids. In Handbook of metaheuristics. 2010; p. 469–96.
[75] Ibrahim M, Nabi S, Baz A, Naveed N, Alhakami H. Towards a task and resource aware task scheduling in Cloud Computing: An experimental comparative evaluation. Int J Networked Distrib Comput. Jun. 2020;8(3):131–8. doi: 10.2991/ijndc.k.200515.003.
[76] Hamid L, Jadoon A, Asghar H. Comparative analysis of task level heuristic scheduling algorithms in cloud computing. J Supercomput. Jul. 2022;78(11):12931–49. doi: 10.1007/s11227-022-04382-x.
[77] Nayak SC, Parida S, Tripathy C, Pattnaik PK. An enhanced deadline constraint based task scheduling mechanism for cloud environment. J King Saud Univ - Comput Inf Sci. Feb. 2022;34(2):282–94. doi: 10.1016/j.jksuci.2018.10.009.
[78] Mangalampalli S, Karri GR, Kumar M, Khalaf OI, Romero CAT, Sahib GMA. DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing. Multimed Tools Appl. Jan. 2024;83(3):8359–87. doi: 10.1007/s11042-023-16008-2.
[79] Mishra R, Gupta M. Cloud scheduling heuristic approaches for load balancing in cloud computing. In 2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023. Institute of Electrical and Electronics Engineers Inc.; 2023. doi: 10.1109/ISCON57294.2023.10112056.
[80] Yao F, Pu C, Zhang Z. Task duplication-based scheduling algorithm for budget-constrained workflows in cloud computing. IEEE Access. 2021;9:37262–72. doi: 10.1109/ACCESS.2021.3063456.
[81] Alhaidari F, Balharith TZ. Enhanced round-robin algorithm in the cloud computing environment for optimal task scheduling. Computers. May 2021;10(5):63. doi: 10.3390/computers10050063.
[82] Mustapha SMFDS, Gupta P. Fault aware task scheduling in cloud using min-min and DBSCAN. Internet Things Cyber-Phys Syst. Jan. 2024;4:68–76. doi: 10.1016/j.iotcps.2023.07.003.
[83] Khan MA. A cost-effective power-aware approach for scheduling cloudlets in cloud computing environments. J Supercomputing. Jan. 2022;78(1):471–96. doi: 10.1007/s11227-021-03894-2.
[84] Alsaidy SA, Abbood AD, Sahib MA. Heuristic initialization of PSO task scheduling algorithm in cloud computing. J King Saud Univ - Comput Inf Sci. Jun. 2022;34(6):2370–82. doi: 10.1016/j.jksuci.2020.11.002.

[85] Prathiba S, Sankar S. An optimal learning-based optimizer for task scheduling and resource utilization in online and offline cloud environment. In IEEE 9th International Conference on Smart Structures and Systems, ICSSS 2023. Institute of Electrical and Electronics Engineers Inc.; 2023. doi: 10.1109/ICSSS58085.2023.10407749.
[86] Yuvaraj N, Karthikeyan T, Praghash K. An improved task allocation scheme in serverless computing using gray wolf optimization (GWO) based reinforcement learning (RIL) approach. Wirel Pers Commun. Apr. 2021;117(3):2403–21. doi: 10.1007/s11277-020-07981-0.
[87] Nanjappan M, Natesan G, Krishnadoss P. An adaptive neuro-fuzzy inference system and black widow optimization approach for optimal resource utilization and task scheduling in a cloud environment. Wirel Pers Commun. Dec. 2021;121(3):1891–916. doi: 10.1007/s11277-021-08744-1.
[88] Tamilarasu P, Singaravel G. Quality of service aware improved coati optimization algorithm for efficient task scheduling in cloud computing environment. J Eng Res (Kuwait). 2024;12(4):768–80. doi: 10.1016/j.jer.2023.09.024.
[89] Chen X, Cheng L, Liu C, Liu Q, Liu J, Mao Y, et al. A WOA-based optimization approach for task scheduling in cloud computing systems. IEEE Systems J. 2020;14(3):3117–28. doi: 10.1109/JSYST.2019.2960088.
[90] Su Y, Bai Z, Xie D. The optimizing resource allocation and task scheduling based on cloud computing and ant colony optimization algorithm. J Ambient Intell Humaniz Comput. 2021;1–9. doi: 10.1007/s12652-021-03445-w.
[91] Alghamdi MI. Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO). Sustainability (Switz). Oct. 2022;14(19):11982. doi: 10.3390/su141911982.
[92] Shrichandran GV, Narayana Tinnaluri VS, Senthil Murugan J, Meeradevi T, Dwivedi VK, Suma Christal Mary S. Hybrid competitive swarm optimization algorithm based scheduling in the cloud computing environment. In Proceedings of the 5th International Conference on Inventive Research in Computing Applications, ICIRCA 2023. Institute of Electrical and Electronics Engineers Inc.; 2023. p. 1013–8. doi: 10.1109/ICIRCA57980.2023.10220842.
[93] Mishra K, Majhi SK. A binary Bird Swarm Optimization based load balancing algorithm for cloud computing environment. Open Comput Sci. Jan. 2021;11(1):146–60. doi: 10.1515/comp-2020-0215.
[94] Paulraj D, Sethukarasi T, Neelakandan S, Prakash M, Baburaj E. An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment. PLoS One. Mar. 2023;18(3):e0282600. doi: 10.1371/journal.pone.0282600.
[95] Hu B, Cao Z, Zhou M. Scheduling real-time parallel applications in cloud to minimize energy consumption. IEEE Trans Cloud Comput. 2022;10(1):662–74. doi: 10.1109/TCC.2019.2956498.
[96] Xu J, Zhang Z, Hu Z, Du L, Cai X. A many-objective optimized task allocation scheduling model in cloud computing. Appl Intell. Jun. 2021;51(6):3293–310. doi: 10.1007/s10489-020-01887-x.
[97] Bandaranayake KMSU, Jayasena KPN, Kumara BTGS. An efficient task scheduling algorithm using total resource execution time aware algorithm in cloud computing. In Proceedings - 2020 IEEE International Conference on Smart Cloud, SmartCloud 2020. Institute of Electrical and Electronics Engineers Inc; Nov. 2020. p. 29–34. doi: 10.1109/SmartCloud49737.2020.00015.
[98] Alsubai S, Garg H, Alqahtani A. A novel hybrid MSA-CSA algorithm for cloud computing task scheduling problems. Symmetry (Basel). Oct. 2023;15(10):1931. doi: 10.3390/sym15101931.
[99] Kaur A, Kaur B. Load balancing optimization based on hybrid Heuristic-Metaheuristic techniques in cloud environment. J King Saud Univ - Comput Inf Sci. Mar. 2022;34(3):813–24. doi: 10.1016/j.jksuci.2019.02.010.
[100] Moazeni A, Khorsand R, Ramezanpour M. Dynamic resource allocation using an adaptive multi-objective teaching-learning based optimization algorithm in cloud. IEEE Access. 2023;11:23407–19. doi: 10.1109/ACCESS.2023.3247639.
[101] Zhao Y, Ye Z, Chen K, Lu Q, Xiang Z. A cloud resource allocation strategy with entry control for multi-priority cloud requests. Arab J Sci Eng. Aug. 2023;48(8):10405–15. doi: 10.1007/s13369-023-07635-w.
[102] Aktan MN, Bulut H. Metaheuristic task scheduling algorithms for cloud computing environments. In Concurrency and computation: practice and experience. John Wiley and Sons Ltd; Apr. 2022. doi: 10.1002/cpe.6513.
[103] Albert P, Nanjappan M. WHOA: Hybrid based task scheduling in cloud computing environment. Wirel Pers Commun. Dec. 2021;121(3):2327–45. doi: 10.1007/s11277-021-08825-1.
[104] Nekooei-Joghdani A, Safi-Esfahani F. Dynamic scheduling of independent tasks in cloud computing applying a new hybrid metaheuristic algorithm including Gabor filter, opposition-based learning, multi-verse optimizer, and multi-tracker optimization algorithms. J Supercomputing. Jan. 2022;78(1):1182–243. doi: 10.1007/s11227-021-03814-4.
[105] Pirozmand P, Hosseinabadi AAR, Farrokhzad M, Sadeghilalimi M, Mirkamali S, Slowik A. Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. Neural Comput Appl. Oct. 2021;33(19):13075–88. doi: 10.1007/s00521-021-06002-w.
[106] Dokeroglu T, Kucukyilmaz T, Talbi EG. Hyper-heuristics: A survey and taxonomy. Comput Ind Eng. Jan. 2024;187:109815. doi: 10.1016/j.cie.2023.109815.
[107] Xiao Q-Z, Zhong J, Feng L, Luo L, Lv J. A cooperative coevolution hyper-heuristic framework for workflow scheduling problem. IEEE Trans Serv Comput. 2019;15(1):150–63. doi: 10.1109/TSC.2019.2923912.
[108] Drake JH, Kheiri A, Özcan E, Burke EK. Recent advances in selection hyper-heuristics. Eur J Oper Res. 2020;285(2):405–28. doi: 10.1016/j.ejor.2019.07.073.
[109] Li C, Wei X, Wang J, Wang S, Zhang S. A review of reinforcement learning based hyper-heuristics. PeerJ Comput Sci. 2024;10:e2141. doi: 10.7717/peerj-cs.2141.

[110] Yin L, Sun C, Gao M, Fang Y, Li M, Zhou F. Hyper-heuristic task scheduling algorithm based on reinforcement learning in cloud computing. Intell Autom Soft Comput. 2023;37(2):1587–608. doi: 10.32604/iasc.2023.039380.
[111] Gupta A, Bhadauria HS, Singh A. Load balancing based hyper heuristic algorithm for cloud task scheduling. J Ambient Intell Human Comput. 2021;12:5845–52. doi: 10.1007/s12652-020-02127-3.
[112] Institute of Electrical and Electronics Engineers, IEEE Computational Intelligence Society, and Victoria University of Wellington, 2019 IEEE Congress on Evolutionary Computation (CEC): 2019 proceedings. doi: 10.1109/CEC.2019.8790220.
[113] Freire DL, Frantz RZ, Roos-Frantz F, Basto-Fernandes V. Queue-priority optimized algorithm: a novel task scheduling for runtime systems of application integration platforms. J Supercomputing. Jan. 2022;78(1):1501–31. doi: 10.1007/s11227-021-03926-x.
[114] Tong Z, Chen H, Liu B, Cai J, Cai S. A novel intelligent hyper-heuristic algorithm for solving optimization problems. J Intell Fuzzy Syst. 2022;42(6):5041–53. doi: 10.3233/JIFS-211250.
[115] Bouazza W, Sallez Y, Trentesaux D. Dynamic scheduling of manufacturing systems: a product-driven approach using hyper-heuristics. Int J Comput Integr Manuf. 2021;34(6):641–65. doi: 10.1080/0951192X.2021.1925969.
[116] Pradhan A, Bisoy SK, Das A. A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. J King Saud bin Abdulaziz University – Comput I Sci. 2022;34(8):4888–901. doi: 10.1016/j.jksuci.2021.01.003.
[117] Sanchez M, Cruz-Duarte JM, Ortiz-Bayliss JC, Ceballos H, Terashima-Marin H, Amaya I. A systematic review of hyper-heuristics on combinatorial optimization problems. IEEE Access. 2020;8:128068–95. doi: 10.1109/ACCESS.2020.3009318.
[118] Vela A, Cruz-Duarte JM, Ortiz-Bayliss JC, Amaya I. Beyond hyper-heuristics: a squared hyper-heuristic model for solving job shop scheduling problems. IEEE Access. 2022;10:43981–4007. doi: 10.1109/ACCESS.2022.3169503.
[119] Du T, Xiao G, Chen J, Zhang C, Sun H, Li W, et al. A combined priority scheduling method for distributed machine learning. EURASIP J Wirel Commun Netw. Dec. 2023;2023(1):45. doi: 10.1186/s13638-023-02253-4.
[120] Alshareef HN. Current development, challenges and future trends in cloud computing: a survey. Int J Adv Comput Sci Appl. 2023;14(3). doi: 10.14569/IJACSA.2023.0140337.