Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2024.050681

REVIEW

Systematic Review: Load Balancing in Cloud Computing by Using Metaheuristic Based Dynamic Algorithms

Darakhshan Syed*, Ghulam Muhammad and Safdar Rizvi

Department of Computer Science, Bahria University, Karachi Campus, Karachi, Pakistan

*Corresponding Author: Darakhshan Syed. Email: darakshansyed.bukc@bahria.edu.pk

Received: 13 February 2024 Accepted: 28 April 2024 Published: 11 July 2024

ABSTRACT

Cloud Computing has the ability to provide on-demand access to a shared resource pool. It has completely changed the way businesses are managed, implement applications, and provide services. The rise in popularity has led to a significant increase in the user demand for services. However, in cloud environments efficient load balancing is essential to ensure optimal performance and resource utilization. This systematic review targets a detailed description of load balancing techniques including static and dynamic load balancing algorithms. Specifically, metaheuristic-based dynamic load balancing algorithms are identified as the optimal solution in case of increased traffic. In a cloud-based context, this paper describes load balancing measurements, including the benefits and drawbacks associated with the selected load balancing techniques. It also summarizes the algorithms based on implementation, time complexity, adaptability, associated issue(s), and targeted QoS parameters. Additionally, the analysis evaluates the tools and instruments utilized in each investigated study. Moreover, comparative analysis among static, traditional dynamic and metaheuristic algorithms based on response time by using the CloudSim simulation tool is also performed. Finally, the key open problems and potential directions for the state-of-the-art metaheuristic-based approaches are also addressed.

KEYWORDS

Cloud computing; load balancing; metaheuristic algorithm; dynamic algorithm; load balancer; QoS

1 Introduction

Cloud Computing has globally evolved as a transformative paradigm. Data is systematically stored, processed, and accessed through the internet as opposed to just one unit of individual hardware or a workplace network [1]. In contrast to conventional techniques that depend on specific hardware components or regional networks, cloud computing makes use of the internet to provide scalable and effective access to a wide variety of services, software programs, and platforms. Large data centers built to optimize operational efficiency house these services [2]. Cloud service providers utilize a variety of models to deliver the services. IaaS (infrastructure as a service), PaaS (platform as a service), and SaaS (software as a service) are the three widely used methods [3,4]. These frameworks are often depicted as pyramids due to the increasing levels of abstraction.

The widespread acceptance of cloud computing in the commercial domain is mostly due to its intuitive user interfaces and strong security protocols. This encompasses services like Amazon, Azure, and Google engine. Other educational and corporate service providers have also gained popularity in recent years by offering the services at reasonable prices while maintaining high levels of competence [5]. The way computational operations have dealt with vast amounts of data has been made possible by cloud services. These services have captured people's attention and supported a variety of financial operations. It has been attracting a significant number of individuals who, upon accepting their requests, utilize its services and subsequently remunerate the appropriate amount for the duration of resource usage [6]. However, Cloud Service Providers (CSPs) face significant challenges in optimizing, scaling, and securing applications while maintaining high user satisfaction levels due to the increasing demand for cloud computing. This is where the load balancer (LB) plays a crucial role. As depicted in Fig. 1, the load balancer is a fully distributed software-defined solution designed to distribute user traffic evenly across multiple back-end services, hereby averting congestion and guaranteeing minimal latency.

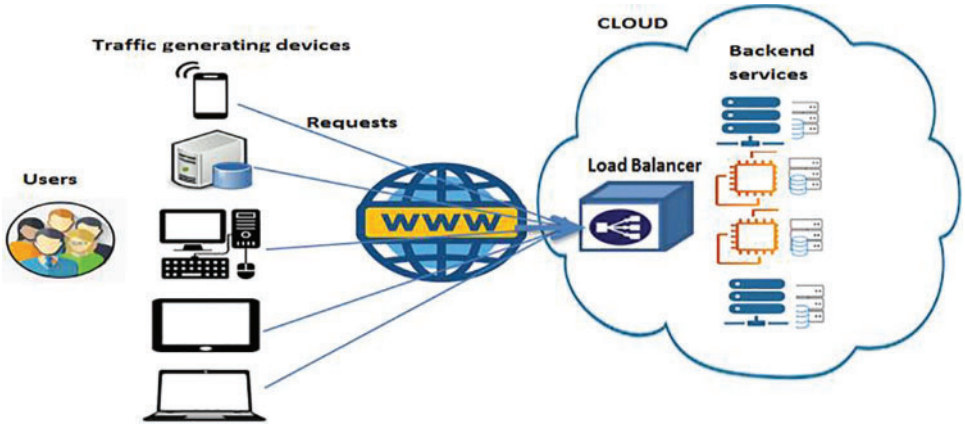

Figure 1: Load balancer in a cloud computing environment

The role of the load balancer is to select the server instance by using an algorithm. The selection of an optimal load balancing algorithm is the aim of researchers due to the heterogeneity of VMs. There are multiple possibilities of mappings, which make the assignment of tasks an NP-hard problem [7,8]. Furthermore, there are several regulatory and compliance considerations that need to be considered when implementing load balancing techniques in cloud computing environments. Data privacy laws ensure that the processed data via a load balancer must comply with security-related rules. The load balancing configurations must support confidentiality and integrity in the case of healthcare cloud-based applications for maximizing patients' satisfaction [9,10]. Financial Services regulations also need improved security; the load balancing algorithms also comply with these policies to secure sensitive financial data [11]. Load balancing methodologies must respect legal constraints associated when the offered application being globally accessible.

The scalability and efficiency of the load balancing techniques in large-scale cloud deployments are critically significant considerations to ensure optimal resource utilization. Scalability considerations involve efficient management incoming requests from multiple data centers, intelligent decision-making capability while considering the real-time information of resources, ability to deal with a sudden spike in traffic, and auto-adjust server instances based on the incoming user requests [12,13]. While efficiency considerations entail preventing server overutilization or underutilization,

continuously monitoring server health, tracking failed nodes, migrating resources, ensuring high availability for critical applications [14], minimizing overhead, and simplifying algorithm complexity. Additionally, high adaptability to server capabilities and user retention is also crucial for dynamic allocation efficiency [15]. Service Level Agreements (SLAs) are used in services-oriented clouds for negotiating services between CSPs and customers [16]. Because the specifications are known prior, static schedulers [17] can perform the scheduling of resources effectively in this situation. In this area of study, heuristic and meta-heuristic schedulers [18–21] are used to identify the best scheduling. Given the enormous search space of potential solutions, meta-heuristics can better serve the goal of arriving at a better answer while examining an extensive search space.

1.1 Research Objectives

The primary objective of this study is to present an overview of load balancing studies conducted in cloud computing that utilize metaheuristic algorithms, along with their evolution. The study examines modern approaches to load balancing and various hybrid techniques, covering the period from 2010 to 2024. The existing load balancing methods are analyzed in a logical structure. The aim is to provide a clear and accurate understanding of the underlying algorithms employed in each approach. To achieve this, it is essential to investigate the research conducted on load balancing in the context of cloud computing, starting from the introduction of static algorithms and progressing to the current state-of-the-art techniques. The prime objectives of this study are summarized as follows:

1. To present a detailed review of the evolution of load balancing algorithms.
2. To highlight the advantages and disadvantages associated with load balancing algorithms.
3. To present a detailed summary of load balancing algorithms.
4. To discuss how metaheuristic algorithms outperform other static and traditional dynamic algorithms.
5. To show a detailed comparative analysis based on trends of the targeted papers and QoS of some state-of-the-art metaheuristic-based load balancing algorithms in a cloud computing environment.
6. To offer recommendations and provide an in-depth analysis based on the reviewed literature.

1.2 Research Motivation

The motivation behind this systematic review is to present a complete evolution of load balancing algorithms along with advantages and disadvantages [22]. To provide assistance to future researchers in selecting appropriate algorithm(s) for enhancing load balancer performance. In order to reduce the implementation and migration cost of the data center (DC) through resource consumption optimization [23,24] and resource efficiency [25–29], load balancing is a fundamental requirement for the effective utilization of cloud computing. Additionally, it contributes in achieving required QoS [30]. To properly exploit the massive scale computer. Additionally, it plays a vital role in achieving optimal hardware performance in data centers. Load balancing is crucial in cloud computing to effectively manage the high volume of user demands. It is worth noting that despite the extensive research conducted on load balancing in the cloud, there is a scarcity of published surveys in this area [12,20,31–33]. The relevant surveys in the area of load balancing in Cloud Computing are [34–37]. These surveys provide only a brief explanation of the working of the load balancing algorithms proposed by researchers [38,39]. Furthermore, many of these review papers have limitations in terms of the factors used to analyze the performance parameters of load balancing. These QoS parameters include response time, migration cost, energy conservation, throughput, convergence, and scalability [40–45]. None of the existing reviews has provided an appropriate classification for classifying various

load balancing strategies. To provide researchers with an operational level understanding of state-of-the-art techniques for load balancing in the cloud environment, the current study aims to conduct a thorough and systematic analysis of these techniques. Even some surveys are targeting a single specific technique [46]. The workings of various load balancing algorithms introduced for cloud-based applications in recent years are clearly and precisely studied in this study.

The rest of the paper comprises of the following sections. The background and organization of the review are covered in Section 2. A detailed literature review is provided in Section 3. In Section 3.1, static load balancing algorithms are presented. In Section 3.2, the details of dynamic algorithms are presented to facilitate load balancing in cloud computing. The summary of the static and dynamic algorithms, along with their respective benefits and drawbacks, is presented in Section 4. Section 5 presents the comparative analysis of the systematically reviewed research papers based on QoS parameters and certain other criteria. Section 6 presents the conclusion to elaborate the effectiveness of the metaheuristic-based algorithms used for load balancing in cloud computing along with the future directions.

2 Background and Review Organization

The survey structure includes a comprehensive list of research publications, sources of data, and investigation criteria that are described in the subsequent paragraphs.

2.1 Research Questions

The following are the main research questions that are examined and eventually support the presented systematic review:

1) What is the purpose of load balancing in cloud computing, and why is it important?
2) What are the primary load-balancing algorithms categories in cloud computing?
3) What is the fundamental concept behind metaheuristic algorithms?
4) What are the potential trade-offs associated with different load balancing techniques in terms of resource utilization, response time, and overall system performance?
5) How would metaheuristic algorithms be used to optimize current cloud computing challenges?
6) What potential advantages and effects might the hybrid metaheuristic algorithms have on the QoS metrics of cloud computing efficiency?

The questions can be addressed by delivering precise and effective Cloud backend services and ensuring optimal load balancing based on the review paper and according to the researcher's guidelines. The conclusive answers are presented below:

1) What is the purpose of load balancing in cloud computing, and why is it important?

As Amazon is using its own load balancer, but still when the user requests are increased exponentially, they are not routed to targets due to high network congestion. Therefore, cloud services need load balancing. The workload can be divided through load balancing in accordance with the resources offered. Its objective is to maintain continuous response by making available and disposing off application versions and making appropriate use of resources in the scenario that any service component fails [47]. Reduced response time for operations and optimal resource utilization boosts device performance at a lower cost attributable to load balancing. It also seeks to assure sustainability and adaptability for operations whose scale will increase soon and require increased services in terms of various resources to assist any given service, as well as to prioritize those operations that must

be done immediately in contrast to others. The capacity of load balancing to reduce energy usage [48,49], prevent bottlenecks, improve scalability [50], offer maintenance, and satisfy QoS requirements is another key reason for adapting it [51].

2) What are the primary load-balancing algorithms categories in cloud computing?

Load balancing in Cloud Computing is divided into two types, i.e., static and dynamic algorithms. Frameworks with limited changes in demand should use static algorithms [52,53]. The traffic is distributed equally among the servers in a static mechanism. This algorithm needs a prior understanding of the system resources. The selection of when to shift the load is not contingent on the system's present condition because the processing units' efficiency will be assessed at the very beginning of execution [22,54]. In continuously changing and distributed infrastructures, dynamic algorithms produce improved outcomes. These techniques offer greater flexibility. Dynamic methods can consider the qualities' variations over time. The main benefit of this approach is that task transfer for execution is dependent on the environment's present state, which can assist to increase system efficiency in terms of QoS parameters [55].

3) What is the fundamental concept behind metaheuristic algorithms?

Metaheuristic algorithms are search techniques created to locate a suitable answer to an optimization challenge that is extensive and challenging to address. In this real-world with scarcity of resources, it is crucial to develop a close-to-optimal strategy based on faulty or insufficient knowledge it can be computational power and time. One of the most significant developments in operational research over the past 20 years has been the development of metaheuristics for resolving these optimization issues [56]. Scholars that have written extensively about numerous applications of metaheuristic methods to deal with non-linear non-convex optimization problems thoroughly examine these approaches. Some NP-hard (nondeterministic polynomial time) problems in evolutionary computation are challenging (i.e., in reasonable run time). Because of this, optimization approaches, iterative approaches, and pure greedy approaches are typically less effective in generating effective responses than metaheuristics [57]. An exceptionally demanding optimization technique must be used to reach global optimality and contribute to an optimal scheduling.

Different fields can benefit greatly from the use of metaheuristics. Multi-objective functions having non-linear requirements with certain limitations are the fundamental building blocks of many optimization challenges [58]. For instance, because they are largely non-linear, most optimization challenges in engineering call for solutions to multi-objective problems. The development of the optimization problem to tackle optimized solution, on the other hand, is difficult when dealing with AI and machine learning problems, which heavily rely on massive datasets. As a result, metaheuristics are essential for assisting in the resolution of real-world challenges that are hard to solve using traditional optimization mechanisms [59,60].

4) What are the potential trade-offs associated with different load balancing techniques in terms of resource utilization, response time, and overall system performance?

The selection among static, traditional dynamic and metaheuristic-based load balancing algorithms involves the potential trade-offs associated with different load balancing techniques in terms of resource utilization, response time, and overall system performance. Following are the comparisons of each technique based on the mentioned parameters.

Static load balancing techniques are easy to implement and take less computation time. These techniques degrade in terms of scalability and availability in dynamic environments. They are not

able to manage unpredictable changes in the incoming user requests therefore they lead to inefficient resource utilization. Additionally, an imbalanced resource distribution is experienced by these techniques as some of the data centers are underutilized and some are over utilized [17]. Under static incoming traffic scenarios, the static algorithms perform well but if they face variations in the traffic then the response time eventually increases because their adaptability to changing environments is very limited. The overall performance of these algorithms is efficient in the case of stable environments. On the other hand, in the case of dynamic environments, the inability to adapt to shifts in server availability or demand might lead to a degradation in the cloud infrastructure's overall performance [61]. Traditional Dynamic load balancing techniques are adaptable to changing user requests and server availability. They can manage unpredictable variations in the incoming user requests therefore they lead to efficient resource utilization [62]. Additionally, a balanced resource distribution is experienced by these techniques as none of the data centers are underutilized or over utilized. They can improve user satisfaction by reallocating the resources depending on the current state of the allocated resources and incoming user requests. This dynamic resource management reduces the response time and improves availability, but it requires continuous monitoring [63]. These algorithms generally experience the overhead associated with dynamic allocation of resources and the run-time assignment can also degrade the performance in terms of response time. It is observed that the overall performance of these algorithms outperforms the static algorithms. But their complex nature still creates other challenges and possible traffic congestion [64].

Metaheuristic-based load balancing techniques are not only highly adaptable to changing user requests and server availability but also offer improved optimization. They can optimally handle challenging resource distribution issues, which improves the utilization of resources. The efficiency of resource distribution can be further improved by considering the configuration setup and coefficient assignments. These artificial intelligence (AI)-based algorithms efficiently map incoming traffic to available resources and ultimately reduce response time to a considerable degree. By adjusting to a variety of dynamic scenarios, handling a sudden spike in traffic, improving convergence, and identifying effective distribution strategies, these techniques can significantly enhance the overall system performance [65].

5) How would metaheuristic algorithms be used to optimize current cloud computing challenges?

The distinctive topic Metaheuristics in Cloud Computing is a collection of several offerings that use advanced stochastic optimization methods including computational mathematics and metaheuristics in order to improve the state-of-the-art of decision-support systems in the context of cloud computing [66–69]. By reducing the number of unused servers, Faragardi et al. [70] also addressed the scheduling of resources for real-time software in cloud data centers. Employing a method based on metaheuristics, the researchers formalized the analyzed challenge as an integer-linear optimization challenge. Yousefipour et al. [71] proposed to optimize the consolidation approach for minimizing the overall quantity of operational physical servers in order to lower power consumption and expenses in cloud data centers. Li [72] implemented methods of dynamic and task-type-dependent servers' speed control that are used to maximize the efficiency of the data center and reduce the power consumption of a data center with diverse workloads. The paper specifically addressed the issue of determining the optimal load distribution and data center speed configuration for various classes of services running on heterogeneous systems with varying capabilities.

6) What potential advantages and effects might the hybrid metaheuristic algorithms have on the QoS metrics of cloud computing efficiency?

Several researchers [73–75] have highlighted the potential advantages and effects that the hybrid metaheuristic algorithms might have on the QoS metrics of cloud computing efficiency. Alboaneen et al. [76] came to the conclusion that employing hybrid metaheuristic approaches can benefit from both techniques. The benefits of the alternative technique can compensate for the weaknesses of the first. The effectiveness of the approach or the rate at which metaheuristic approaches approach convergence can both be improved by hybrid metaheuristic approaches. Rahman et al. 's [77] dynamic hybrid heuristics algorithm makes use of the dynamic characteristics relying on metaheuristics in addition to the load level improvement capabilities of metaheuristic based methods. Javanmrdi et al. [78] provided a hybrid approach that accounts for task scheduling while drastically reducing on overall response time and costs. The goal of this suggestion is to use fuzzy theory to improve the conventional Genetic algorithm and lower the number of iterations required to create a swarm. Simulations show that the presented technique is efficient in terms of execution time, computation complexity, and the degree of imbalance.

2.2 Search Engines Selection Criteria

This study focused on metaheuristic-based algorithms and provided a thorough explanation of load balancing in a cloud setting. The following keywords were used: cloud computing, load balancing, static algorithms, dynamic algorithms, metaheuristic algorithms, QoS parameters, and issues with cloud load balancing. The details of the search engines are presented in Table 1.

Table 1: Selected search engines for the review

Journal	Link (Accessed date)
IEEE Xplore	https://ieeexplore.ieee.org/ (accessed on 17 Mar 2024)
ACM	https://acm.org/ (accessed on 10 Feb 2024)
Academia	https://academia.edu/ (accessed on 30 Jan 2024)
Science direct	https://sciencedirect.com (accessed on 15 Mar 2024)
Taylor and Francis	https://www.taylorandfrancis.com (accessed on 15 Jan 2024)
Hindawi	https://www.hindawi.com (accessed on 8 Feb 2024)
Elsevier	https://www.elsevier.com (accessed on 16 Mar 2024)
Springer	https://springer.com (accessed on 16 Mar 2024)

2.3 Data Collections Criteria

A variety of data sources were considered for this study. The primary sources for gathering corresponding research papers were Google Scholar, research papers, books, and webpages. The percentages of different published studies that were examined during the period from 2010 to 2024, based on year and sources are also shown in Figs. 2 and 3.

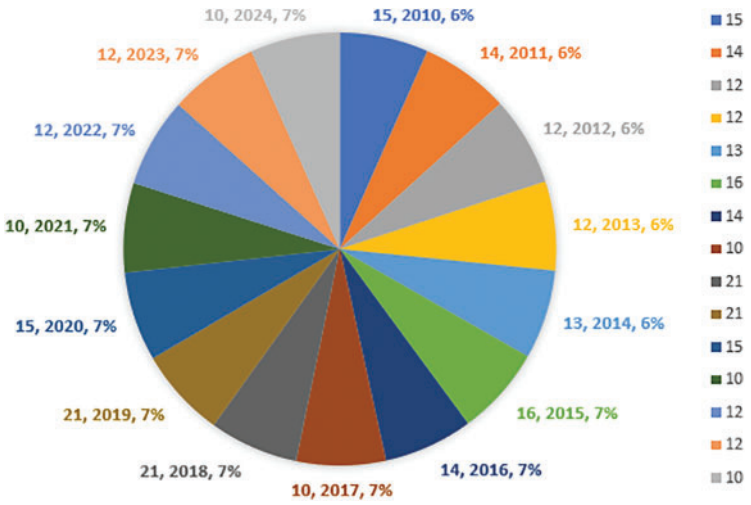

Figure 2: The number (with percentage) of reviewed research papers from 2010 to 2024

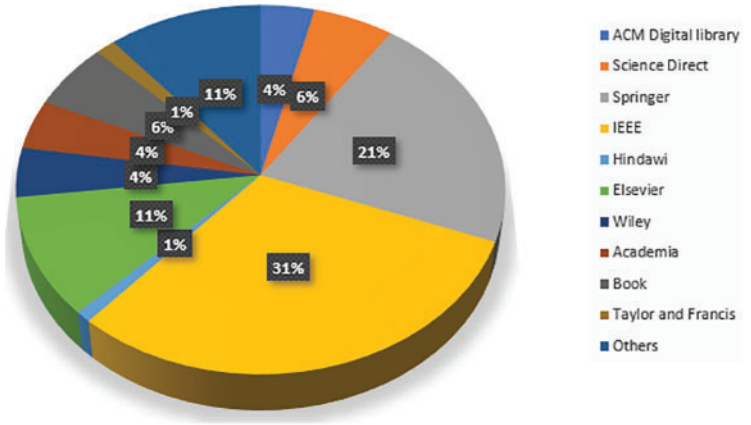

Figure 3: The percentage of sources reviewed from 2010 to 2024

2.4 Inclusion and Exclusion Criteria

The gathered papers for research were then required to adhere to standards of assessment for inclusion and exclusion. Any manuscripts were excluded after the initial abstract assessments. An Excel sheet is used to organize the data extracted from the relevant papers and generate summary statistics. The main criteria for inclusion and exclusion are listed in Table 2.

Table 2: Decision of inclusion and exclusion criterion for paper selection

The rules followed for selecting research content	
Inclusion	A professionally demonstrated research paper. Comprehensive research conducted in load balancing in cloud computing.

(Continued)

Table 2 (continued)

The rules followed for selecting research content	
	A comprehensive explanation of both static and dynamic load balancing methods for cloud computing. Research introducing the idea of load balancing algorithms based on metaheuristics that improves the QoS parameters. A detailed paper is written in the English Language.
Exclusion	A research study that does not highlight the challenge of load balancing. A study that focuses on cloud computing issues other than scheduling tasks and utilization of resources. A paper not written in English Language. A research study that does not highlight the challenge of load balancing.

3 Review of Static and Dynamic Load Balancing Algorithms for Cloud Computing

The need for online provisioning of services has grown due to the recent evolution of cloud computing. Consequently, there is now a higher demand for cloud servers and more traffic that needs to be managed. A better load balancer that spreads the incoming traffic across several instances is required to handle the ongoing increase in incoming tasks [14]. This lessens the possibility of performance issues, reduces the amount of time it takes for a task to respond, and improves resource efficiency [79]. Many cloud-based businesses use both static and dynamic load balancing techniques. Based on performance indicators, Sowjanya et al. [80] provided a critical analysis of different load balancing strategies. Hota et al. [81] provided a thorough explanation and comparative analysis of several load balancing techniques. The paper also discusses the benefits and drawbacks of each of the most advanced methods, which may inspire academics to develop load balancing algorithms more extensively. The following specific criterions are used to select the load balancing techniques for the systematic review:

- Most targeted research papers contain relevance to load balancing in cloud computing published recently. Older papers are only considered if they offer underlying ideas or analogies, contingent on the review's scope.
- The load balancing algorithms must have a diverse range of quality-of-service parameters so that the review can support future researchers in taking advantage of the complementary benefits of multiple algorithms.
- Algorithms that have substantial amount of research literature are more likely to be included in the review. As they provide ample information for comparative analysis.
- Each technique has distinct applications and challenges and can be applied to various cloud computing load balancing scenarios. In that way researchers can have a clear understanding of various algorithms that may perform differently under different circumstances.

The highlighted approaches typically revolve around dynamic user requests, scalability, availability, fault tolerance, reducing makespan, optimal cost, and efficient energy in research publications that highlight the problem of load balancing in cloud infrastructure. Therefore, the load balancing techniques that address the mentioned critical challenges posed by cloud computing environments can ultimately support improving resource utilization, reducing response time, and the effectiveness of

costs by solving these basic issues and facilitating the hassle-free and effective provisioning of cloud computing services.

According to the published literature it became clear that only a small number of research studies addressed load balancing approaches based on metaheuristic methodologies. Now, a thorough review of static and dynamic load balancing algorithms is provided in the following section.

3.1 Static Load Balancing Algorithms for Cloud Computing

Frameworks with limited changes in demand should use static algorithms [82]. The traffic is distributed equally among the servers in a static mechanism. This algorithm needs prior understanding of the system resources. The selection of when to shift the load is not contingent on the system's present condition because the processing units' efficiency will be assessed at the very beginning of execution [17,54,83]. The round-robin strategy is used by the round-robin (R-R) load balancing technique [84] to distribute workloads. It chooses the first data center randomly and then evenly distributes jobs across the remaining data centers. When the load is evenly distributed among methods, this has the benefit of reduced response time. On the other hand, various procedures have varied processing speeds; some nodes might always be substantially busy while others might remain underutilized. Consequently, weighted round robin was developed to address a few significant problems with the R-R algorithm. Each of the instances in this technique was given a weight, and the instance with the highest weight obtained additional connections. The DCs will receive balanced traffic if every weight has the same value [85]. As a method of attempting to keep each node active, Wang et al. [86] recommended the opportunistic load balancing algorithm. As a result, it does not consider the current tasks of each machine. Irrespective of the data center's current traffic load, the algorithm distributes incomplete tasks to accessible nodes randomly. Although some DCs are available, bottlenecks will develop since the proposed technique does not account for the node's execution time, which causes the task to be executed more slowly. In a survey, Aslam et al. [87] discussed Min-Min and Max-Min algorithms for load balancing in cloud computing.

All the static load balancing algorithms have the problem that jobs cannot be transferred to another machine to balance the current incoming traffic while they are being executed. They can only be transferred after they have been assigned to the processors or VMs. Most of the time, these approaches handle heterogeneity by manually adjusting the environment. As a result, the data centers are first classified according to their static characteristics, which include processing power, memory, and storage capacity. Due to this manual initial configurational setup, they can not adjust to dynamic variations, which means that regular manual modifications may be needed to sustain performance in a heterogeneous environment. Because the nature of the cloud is dynamic and unconstrained, with high user retention, it is necessary to build algorithms that respond to the actual present system state in order to make decisions on the transfer of loads..

3.2 Dynamic Load Balancing Algorithms for Cloud Computing

In continuously changing and distributed infrastructures, dynamic algorithms produce improved outcomes. These techniques offer greater flexibility. Dynamic methods can take into account the variations of qualities over time [88]. The main benefit of this approach is that task transfer for execution is dependent on the present state of the environment, which can help increase system efficiency in terms of QoS parameters [55].

3.2.1 Traditional Dynamic Load Balancing Algorithms for Cloud Computing

Researchers declare [55,87] the Throttled Load Balancing algorithm as an appropriate approach for VMs. The load balancer ensures that the collection of functional VMs in the system is kept up to date. Whenever a connection request is made, the load balancer scans the referencing table. If a VM is readily available, is chosen to complete the task. Another variant of the Dynamic Load Balancer is the "Equally Spread Current Execution Load Balancing Algorithm" [89,90]. This algorithm keeps track of all active VMs and their workloads. Whenever a request comes in, the load balancer searches the list of VMs and assigns the request to a specific VM that can manage it. This approach evenly distributes the load among all VMs. Another dynamic load balancing algorithm is Random Sampling [90]. Domanal et al. [91] presented dynamic algorithm for load balancing named Modified Throttled, which focuses on optimally distributing workloads to available VMs. These traditional dynamic load balancing algorithms including least connections (LC), weighted least connections (WLC) and Reduced Response Time (RRT), outperform the static load balancing algorithms. However, there is still a need to further improve the load balancer's quality of service parameters for better performance.

Traditional dynamic load balancing techniques keep track of the available data centers' processing power, busy VMs and storage capacity at runtime. In this way, they assign incoming traffic to data centers with improved resource utilization. These tasks, including VM assignment, also depend on the past outcomes and learnings from globally diverse environments. Therefore, they efficiently manage the heterogeneity of cloud environments. Furthermore, due to feedback-based learning they take time to decide on the optimal solution and may experience failure in complex spaces [34].

3.2.2 Metaheuristic Based Dynamic Load Balancing Algorithms for Cloud Computing

Contrary to traditional dynamic load balancing techniques, Metaheuristic-based load balancing algorithms are the best option for dealing with the heterogeneity of cloud environment. Their properties that provide a global optimization, customization, high adaptability and efficient response to varying capabilities and load variations make them a robust solution [92,93].

A PSO-based rescheduling strategy for cloud-based systems was offered by Pandey et al. [94] and it considers the expenses of both computing and data traffic. The computation and connectivity costs of a workflow application are changed to conduct experiments. Wu et al. [95] suggested an updated discrete-PSO scheduling technique that divides workloads across cloud services while taking network connection and computing costs into account. Zamfirache et al. [96] proposed two types of schedulers: evolutionary and ant-based. The choice of the most appropriate operators is examined for each one, as well as various memory-based processes, to evaluate how well they can adapt to the dynamic cloud environment. According to Nishant et al. [97], ant colony optimization (ACO) is improved and implemented from the perspective of cloud or grid network architectures with the primary goal of managing the load of data centers. According to Csorba et al.'s [98] hypothesis, implementing self-organizing techniques for system reconfiguration can improve such frameworks' manageability and capacity for change. Sesum-Cavic et al. [99] described the preliminary results of a unique strategy to efficiently manage the incoming load inspired by Artificial bee social behavior. Considering load balancing to be a multi-objective challenge [100], Kołodziej et al. [101] designed a hierarchical based genetic lolad balancer. Eventually it improves the response time and makespan.

Zhang et al. [102] utilized Ant colony and Complex system-Theory. This research utilizes several of the elements of the older Ant Colony approaches, which were developed to provide resource provisioning in cloud environments, are built into the suggested method. This method also considers the characteristics of Complex Networks. Hu et al. [103] described a GA based scheduling technique

for balancing the load of VM services. This approach evaluates the impact ahead by using past statistics and the contemporary state of the system, and then can choose the least-effective alternative, ensuring the desired load balancing and reducing or avoiding dynamic transfer. Ge et al. [104] offered a new scheduler that evaluates the full set of jobs in the work queue before making a scheduling choice. The novel scheduling algorithm uses a genetic algorithm as its optimum mechanism. Accessing and scheduling cloud online services is always a difficult task, according to Mukherjee et al. [105].

An expedient method for resolving grid task scheduling issues was presented by Chen et al. [106] in their conventional PSO metaheuristic technique. Rahman et al.'s [77] dynamic hybrid heuristics algorithm makes use of the dynamic characteristics of approaches relying on metaheuristics in addition to the load level improvement capabilities of metaheuristic metaheuristic-basedcu and Frincu et al. [107] described an approach for achieving service high-availability and ability tolerate faults while lowering application costs and maximizing resources load. You et al. [108] presented Ursa to provide time-responsive distribution alternatives for computing and enables cost-effective task scheduling while scalable to large systems. In order to solve the load balancing problem in cloud services, Zhu et al. [109] first demonstrated how multi-agent genetic techniques outperform traditional GA. Researchers next develop a load balancing system based on virtualization resource tracking and management. Xu et al. [110] proposed a novel data allocation balancing model to improve cloud performance and scalability in data-intensive scenarios like distributed data mining. The optimum placements are chosen for a scheduling methodology developed by Paul et al. [111] using the Lexi-search strategy. Task scheduling has been considered as a general allocation issue to determine the lowest cost. To construct a cost matrix, a probabilistic component based on some of the most important parameters of efficient scheduling algorithms, such as task entry, task holding duration, and the most crucial task computation in a system, is employed. Zhang et al. [112] proposed an ant colony and genetic algorithm-based adaptive combination. The Genetic Algorithm's preliminary response was converted into the pheromone initial quantity that the ACO algorithm was needed to achieve the best outcome. The time restricted scheduling challenge was addressed by Jian et al. [113] using a Simulated Annealing method inspired by heuristics. Processing of operations and data transfer make up the scheduling cost.

Kim et al. [114] presented a binary artificial bee colony approach for grid computing to intelligently manage task scheduling. Fan et al. [115] suggested a load balancing technique based on simulated annealing. In a cloud computing framework, the goal is to tackle the resource allocation and scheduling issue. Lagaros [116] discussed how parallel processing power can be used in metaheuristic optimization by employing the candidate solutions handling evolution initiatives method's physical parallelization functionality, as well as in repetitive structural analyses required for assessing behavioral restrictions and estimating objective functions. Rana et al. [117] regarded load balancing to be a significant concern. The study provides an overview of numerous available strategies that can be used to improve resource use. Farahnakian et al. [118] discussed the key difficulty in cloud-based technology is to minimize data center energy consumption while meeting QoS standards. Raju et al. [119] addressed numerous techniques to energy conservation, each of which has its own set of drawbacks, for instance time-complexity and convergence delay. To allocate resources and conserve energy, an effective algorithm is created by combining the features of echo-localizing and hibernating. Javanmrdi et al. [78] provided a hybrid approach to task scheduling that accounts for task scheduling while drastically reducing on overall response time and costs. The goal of this suggestion is to use fuzzy theory to improve the conventional Genetic algorithm and lower the number of iterations required to create a swarm. Simulations show that the presented technique is efficient in terms of execution time, computation complexity, and degree of Imbalance. Xue et al. [120] presented the Ant Colony

Optimizer to achieve this task scheduling. The proposed algorithm can adapt to a changing cloud infrastructure. It will not only reduce the time it takes to schedule tasks, but it will also keep the load balance of virtual machines in the data center. Overall incoming traffic balancing, throughput, and makespan are optimized by using Particle Swarm Optimization (PSO) by Chitra et al. [121]; attempt to find an optimal workflow schedule. The recommended algorithm's usefulness is demonstrated by experimental findings. With a larger number of workloads, the proposed technique is more practical as the environment of the cloud is dynamic by nature. Kalra et al. [122] identified scheduling as being among the relevant research concerns that must be rectified if significant performance is to be achieved. According to studies, metaheuristic-based strategies have been shown to generate near-optimal results for such issues in an acceptable amount of duration.

Wen et al. [123] introduced a novel scalable VM reallocation technique centered on the Ant Colony Optimization metaheuristic technique. Experiments reveal that this suggested approach improves conventional rescheduling techniques in terms of attaining system load balancing, lowering the frequency of reallocations, and preserving required quality of service standards. Cho et al. [124] developed ACO with particle swarm optimization (ACO PS), which combines ACO and PSO to tackle the VM scheduling dilemma. ACO PS forecasts using past data despite extra job knowledge, the strain of new incoming data requests to adjust to dynamic contexts ACO PS additionally eliminates queries that cannot be fulfilled before scheduling to reduce the scheduling-computational effort of the procedure. The suggested approach outperforms the alternatives, according to experimental evaluations, in maintaining load balancing in a dynamic environment. Shojafar et al. [125] introduced FUGE, a hybrid method that uses fuzzy logic and GA to balance loads effectively while considering cost and overall response time. The researchers enhance the conventional genetic algorithm and use fuzzy theory to construct an improved technique to optimize GA efficiency in terms of computation time. Sliwko et al. [126] provided a conceptual framework of cloud resources allocation, taking into account numerous types of resources as well as service migration costs. Dasgupta et al. [127] suggested a GA-inspired algorithm that strives to maintain the load on the cloud infrastructure while attempting to reduce the make span of a certain collection of incoming requests. Artificial Bee Colonies, Ant Colony Systems, Genetic, and Particle Swarm Optimization algorithms are all included in the presented survey by Farrag et al. [128]. The research paper also suggests that an ALO-based cloud computing infrastructure be implemented as an efficient technique that will provide better load balancing results.

Davydov et al. [129] considered the server load balancing challenge that arises from cloud computing's ideal web hosting. Researchers devised a VNS-based methodology. The algorithm's effectiveness is demonstrated through computational studies on real-world and dynamically created test cases. Beegom et al. [130] used a genetic algorithm methodology to concurrently optimize the two main purposes, i.e., makespan and cost. Pan et al. [131] recommended using an improved particle technique. This method considers the features of complex networks to produce an efficient resource job distribution mechanism. The simulation results showed that this approach may enhance resource usage and cloud task scheduling. Aslanzadeh et al. [132] provided a unique load balancing methodology inspired by the endocrine technique, which is affected by the regulatory behavior of the hormonal mechanism. Alboaneen et al. [76] came to the conclusion that employing hybrid metaheuristic approaches can benefit from both techniques. Gasior et al. [133] presented a completely online resource allocator based on the Sandpile CA model for dynamic load balancing and reallocation of resources. When a given VM discovers any imbalances, it sends out an unending stream of jobs to his neighbors, which can spread across the system until a new sense of balance is reached.

Masdari et al. [134] carried out a critical investigation of workflow rescheduling methods for cloud data centers. The Memetic algorithm, proposed by Sabar et al. [135], is a hybrid approach that fuses population-based iterative and incremental model with local search technique. The topic of load balancing and the corresponding future developments are highlighted by Milani et al. [34]. The techniques for load balancing that have been reviewed so far have been systematically reviewed. Alboaneen et al. [136] minimized energy consumption and service level agreement (SLA) violations by tackling the VMP challenge with a glowworm swarm optimization (GSO) technique. Simulation findings indicate that the GSO-VMP strategy outperforms currently used conventional metaheuristic-based approaches. According to Madni et al. [137], for resource optimization in IaaS cloud-based solutions, a number of scheduling strategies are already available, though opinions on how well they function are divided. In an initiative to shorten the makespan and use resources as economically as possible. Rajput et al. [138] developed a genetic algorithm-based enhanced load balanced min-min (ILBMM) technique (GA). The CloudSim application was used, and the simulated outcomes demonstrate that it outperforms the existing strategies on with the same objectives.

The goal of Acharya et al. [139] to perform effective resource scheduling algorithm in the cloud. This method develops a suitable resources job distribution framework by considering the special qualities of complex networks. Agnihotri et al. [140] worked on reducing dirty storage during virtual machine live transfer. The data center environment under consideration is a heterogeneous one, with physical hosts with varying configurations. Dave et al. [141] has proposed PSO based algorithm to figure out a better way to the dilemma of cloud computing resource distribution and task scheduling. Gamal et al. [142] presented the load balancing technique Hybrid AB (Artificial Bee) and Ant Colony optimization (ACO) is proposed in this research. It is built on fusing important traits of Ant Colony Optimization, like discovering effective alternatives quickly, with those of the Artificial Bee Colonies technique, like their collaborative social behavior. To address the challenge of load-balancing in Cloud Oriented Infrastructure, a hybrid method was suggested by Mousavi et al. [143]. Combining Teaching-Learning-Based Optimization (TLBO) and GWO has been proposed, it can assist with maximizing performance by distributing load among virtual servers and avoiding the difficulty of capturing into a local ideal solution. Rjoub et al. [144] discussed that the effective loads distribution in Cloud systems is recently a highlighted issue to solve because it is practically NP-complete. The key contribution of the presented algorithm is to optimize the makespan. Gupta et al. [145] reduced the makespan, computational cost, and enhances load balancing by incorporating the concept of ACO to address the job scheduling dilemma in a cloud domain. The recommended load balancing ant colony optimization technique (LB-ACO) shows better performance than the NSGA-II approaches in terms of task scheduling and makespan, according to a comparison. The simulations were carried out using CloudSim Toolkit.

Thakur et al. [146] surveyed the problem of cloud computing resource usage. The investigations found that load balancing helps maximize resource consumption and achieve targeted parameters of service quality in the cloud by using effective resource allocations and traffic redistribution algorithms during schedule time and real-time execution. Nilesh et al. [147] explored and analyzed the usage of swarm intelligence strategy Ant colony systems. Ant colony optimization as inspiration for building a load balancing strategy in cloud environments is presented in this paper. Lagwal et al. [148] described a GA-based solution for load balancing in cloud environment. Deepa et al. [17] provided an in-depth comparison of static and dynamic load balancing strategies currently used in cloud service. Tripathi et al. [149] presented a load balancing method called a hybrid strategy for optimization.

Hanine [150] addressed load balancing as one of the most difficult challenges facing VMs in a server. It requires controlling the burden of each VM in order to improve service quality

(QoS). The study presents many alternatives in the hopes of providing a high quality of service. Belgacem et al. [151] depicted the characteristics of jobs and established a classification of resource transfer scheduling methodologies. The proposed classification methods in this study are based on several types of cloud-based task scheduling. Gohil et al. [152] addressed GWO, a relatively recent method inspired by grey wolf social hierarchy and hunting behavior. An improved GWO (iGWO) is suggested in this work, with an emphasis on the need for a reasonable balance between exploration and extraction. Outcomes from simulations focusing on benchmarks for exploitation and exploration as well as the difficulty of cloud task scheduling, show that iGWO outperforms standard GWO, HS, ABC, and Optimization methods in terms of efficacy, cost, and stability. Dam et al. [153] defined computational intelligence (CI) as the study of developing bio-inspired artificial entities to determine the most likely optimal solution. The study compares the suggested method to several current techniques such as GA-GEL, GA, SHC, and FCFS, and finds that it surpasses other algorithms and meets the consumer's QoS requirements. Garg et al. [154] provided a review of metaheuristic strategies. Zhou et al. [155] presented a hybrid glowworm swarm optimization. The proposed HGSO minimizes unnecessary computation, relies on GSO initialization, speeds convergence, and makes it easier to escape local optima.

Luo et al. [156] offered an enhanced particle swarm optimization approach based on adaptable weights to address the peculiarities of task assignment. The findings of this research reveal that, under the identical settings, the upgraded_PSO technique outperforms the standard PSO method in terms of resource utilization, efficiency, and job completion duration. Alazzam et al. [157], in a research study, suggested a new algorithm named Water Flow Algorithm (WFA) for load balancing. To design an effective load balancer, the suggested approach mimics the movement of water. The suggested WFA-LB algorithm was compared to GA, Round Robin, and Min Min techniques in terms of job migration, average speed, and the number of tasks entertained per unit time. Golchi et al. [158] provided a hybrid algorithm inspired by firefly and enhanced PSO methods for achieving a better average demand for providing improved quality metrics, i.e., effective task responsiveness. Jana et al. [159] have suggested a Modified PSO technique that focuses on two key metrics in cloud scheduler: average scheduling length and successful execution ratio. This research not only presents a modified version but also performs a comparative analysis with Min-Minimum, Max-Min, and Basic PSO techniques. According to simulation results, the Modified PSO approach outperforms the mentioned methods.

Farrag et al. [160] also focused on resolving load balancing challenges. The study discussed the application of metahueristic algorithms named Ant-Lion and Grey wolf optimization in load balancing in the Cloud based infrastructure. The research also compares the performance to those of two well-known swarm algorithms: PSO and Firefly Algorithm (FFA). Mansouri et al. [161] achieved better load balancing by integrating improved PSO with a Fuzzy framework. The proposed FMPSO algorithm is analyzed using CloudSim. The simulation results present that the proposed strategy works better than the alternatives in terms of makepan, efficiency, imbalance degree, throughput, and task completion time. Li et al. [162] proposed an innovative resource provisioning strategy for placing virtualized resources on the physical server cluster in a reasonable manner. Afzal et al. [163] presented a taxonomical classification of load balancing algorithms in cloud based environment. Kaur et al. [164] suggested ACOhm to improve makespan and cost. The two optimization load balancing techniques have been analyzed and compared to see which one is better for the suggested Hybrid approach based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (HDD-PLB) architecture. If heuristic techniques are not integrated with other heuristic or meta-heuristic strategies, Mapetu et al. [165] claimed that the ideal solution will not be obtained. An improved_PSOapproach has been suggested by Valarmathi et al. [166] for boosting job scheduling performance. In this research,

a modified PSO based on data localization is used to resolve the inertia weight allocation problem in the current PSO algorithm for workload rescheduling. The RTPSOB algorithm, which combines the RTPSO and the Bat algorithm, also increases efficiency. FIMPSO is a firefly-improved multi-objective PSO algorithm that was proposed as a hybrid technique by Devaraj et al. [167]. The solution space is reduced using the Firefly (FF) approach, and the increased response is located using the IMPSO technique. The particle with the shortest point-to-line distance is chosen using the IMPSO technique. The simulation outcomes demonstrate that the suggested FIMPSO model worked well when compared to other methods. Agarwal et al. [168] discussed that there are many issues that needs to be address by a Cloud Service Provider. An efficient load balancing method is proposed in this study to decrease performance parameters like makespan and improve the effectiveness of cloud-based solutions.

Junaid et al. [169] showed that metaheuristic-based load balancing strategies offered better alternatives for appropriate resource scheduling and transfer of tasks among the VMs. However, most existing techniques only analyze a few or a few QoS indicators, ignoring a number of critical parameters. Combining these methodologies with machine learning techniques improves their performance effectiveness even further. Bhushan [170] recommended ACO for task scheduling and FIFO and Round Robin for processor scheduling. The proposed technique targets makespan and response time as quality metrics.

A new hybrid model is developed by Junaid et al. [171] that classifies the quantity of files in the cloud using file type formatting. Audio, video, text maps, and photographs are just a few of the media types that can be classified using Support Vector Machine on the cloud. The following metrics are assessed in the cloud: SLA violations, relocation time, utilization, operational costs, and optimization time.

Saber et al. [172] offered three significant upgrades to overcome this load balancing challenge. They first introduce a heterogeneous initialized load balancing technique to conduct a suitable job scheduling procedure that improves the makespan in the case of homogeneous or heterogeneous components and provides a route to accomplish adequate load variation. Second, it provided a hybrid load balancing based on genetic algorithm that combines HILB with genetic technique. Third, a newly developed fitness function that minimizes load variance serves as the foundation for GA. The proposed methodology is also simulated. Balaji et al. [173] discussed how every data center produces a large carbon footprint as a result of its extensive energy demand, and consequently has negative environmental consequences. This work used the adaptive cat swarm optimization (ACSO) algorithm to design a load balancing mechanism to address scalability issues. Muteeh et al. [174] offered a cloud solution based on the Multi-resource_LB Algorithm. Singh et al. [175] approached the task to resources mapping by using a crow search inspired balancing strategy to facilitate better utilization of resources. The proposed approach outperforms the traditional algorithm and was judged to be the most efficient load balancing method.

Singh et al. [92] presented a review focused on several nature-inspired optimization techniques and evaluations based on particular traits that influence the effectiveness and efficiency of their implementation to schedule various jobs in the cloud. This study is based on scheduling parameters like makespan and resource utilization cost. Kaviarasan et al. [176] offered improved load-balancing for each node in the cloud by using an operator-based monarch butterfly optimization and migration process optimizations. Because the proposed study uses population-based search instead of single-solution search strategies, it explores more promising areas of the search space and has a higher exploration rate. Throughout the exploration and exploitation phase, the change in convergence is

shown to be consistently maintained. It outperforms other metaheuristic algorithms in terms of throughput, response time, migration time, fault tolerance and energy consumption.

Elmagzoub et al. [36] presented another review of swarm intelligence-inspired load balancing algorithms. The survey aimed to provide applicability areas, and highly targeted challenges (along with developments) are examined. In addition, the service quality metrics such as average response duration, DC processing duration, and other quality metrics like throughput, reliability, makespan, energy, etc., have been analyzed. Houssein et al. [177] presented a thorough investigation. Still more research is required to clearly identify the research gap. Thakur et al. [178] suggested RAFL, a composite metaheuristic-based optimal resource distribution technique. A phasor particle swarm optimization and dragonfly algorithm-based hybrid optimization method dubbed PPSO-DA is employed in the conceptual methodology to develop an effective resource provisioning plan for balancing the load optimally. The Grey Wolf Optimization algorithm was utilized by Sefati et al. [179] to ensure optimal load balancing depending on the resource reliability competence. Using a new strategy for improving VMs' energy consumption and processing duration incorporated in the migration dilemma was presented by Xu et al. [180]. This article suggests a method based on the GA and PSO methods as it is one of the popular NP-hard problems. To get over PSO techniques' drawbacks—poor convergence and delayed global optimal solutions—the hybrid algorithm makes use of a GA. Al-Wesabi et al. [181] introduced a novel integrated metaheuristics for energy efficiency resource allocation named HMEERA. For efficient utilization of resources, the HMEERA model combines the Group_Teaching Optimization with the Rat Swarm Optimizer algorithm. Al Reshan et al. [8] combined PSO and GWO to improve the distribution of resources and load balancer's performance with optimal convergence. A combined fuzzy particle swarm optimization and genetic algorithm (FPSO-GA) was created by Mirmohseni et al. [182] by combining a fuzzy particle swarm optimization technique.

Suresh et al. [183] suggested an improved variant of the CSO technique used by the controller to choose the optimal baseband and remote radio head combinations after examining the QoS data from the existing BB-RR Head arrangements. Prabhakara et al. [184] provided a procedure relying on a hybrid support and load balancing framework that optimizes the utilization of VMs with comparable load distributions. Gabhane et al. [185] combined ACO with the Tabu search method to create an innovative hybrid solution. Swarm and Kubernetes are also introduced to disperse tasks across numerous data centers while keeping in mind that no hubs should be overloaded with the incoming requests [186]. The Google Lab engineers developed a division called Kubernetes to manage containerized apps in a variety of circumstances [187]. Metaheuristic load balancing algorithms, as compared to conventional dynamic algorithms, could provide more adaptable and flexible load balancing. The research of metaheuristics-based approaches is motivated by a compelling idea to use "collaborative intelligence". In an environment, SI is dispersed, synchronized, and decentralized [188]. The problem of managing load on cloud infrastructures can be addressed by adopting metaheuristic-based techniques for load balancing. An efficient solution to the high availability problem in cloud computing has been found leveraging hybrid metaheuristic approaches [189]. Kumar et al.'s [190] innovative hybrid metaheuristic approach was inspired by the Cat and Mouse-Based Optimizer algorithm (CMOA) and the classic Golden Eagle Optimizer. The suggested technique improves convergence and optimizes the load balancer's QoS parameters by taking into consideration response time, throughput, and server capacity. Sumathi et al. [75] additionally aimed to enhance the load balancer's performance with the integration of ACO and Harries Hawks Optimization algorithms. The overall efficiency of the hybridized load balancing (HLB) algorithm is assessed based on makespan, turnaround, waiting, execution, and response times. Geetha et al. [191] also added to further improve

the load balancer's performance by targeting various QoS parameters at the same time. Apart from this recently various researchers [192–194] have proposed hybrid metaheuristic based algorithms to further optimize the QoS parameters.

4 Summary of the Discussed Static and Dynamic Algorithms

A summary of Static, Traditional dynamic and Metaheuristic based load balancing algorithms is presented in Tables 3–5, respectively.

Table 3: Summary of literature review based on static load balancing algorithms

Scheduling algorithms	Author(s) (Reference)	Advantages	Disadvantages
Static algorithms	Deepa et al. [17]	At the time of compiling, a choice is taken regarding load balancing. Spreads out the server's traffic equally.	Constrained to environments with minor fluctuations in load. Lack the capacity to deal with real time variations in traffic at runtime.
Min-Min	Deepa et al. [17]	Shortest time value for completion. It produces the best results when there are smaller jobs.	Starvation. Variation in the machines and tasks cannot be anticipated.
Max-Min	Deepa et al. [17]	It works more conveniently because the requirements are known in advance.	The process of finishing the task is lengthy.
Round-Robin	Rahmawan et al. [61]	Fairness works better for brief CPU bursts; Fixed time quantum; Simple to grasp. In addition, it can also use priority.	Larger jobs require a lot of time. Short quantum time leads to more contact switching. To produce high performance, the task should be the same.
Weighted round-robin	Devi et al. [195]	Takes care of the capacity of the servers in the group.	Increases the processing time.
Randomized assignment	Rahmawan et al. [61]	Simple to implement. Better stability then round robin.	Can lead to overloading of one server while under-utilization of others.
Resource-aware min-min algorithm	Ali et al. [196]	Improves resource utilization. Improves makespan.	Limited performance incoming traffic.

(Continued)

Table 3 (continued)

Scheduling algorithms	Author(s) (Reference)	Advantages	Disadvantages
Max-min and round-robin algorithm	Moses et al. [197]	Reduced response time. Reduced cost	Performance degrades in case of heterogeneous environment.
Maximum-average algorithm	Ibrahim et al. [198]	Improve resource utilization. Reduced cost.	Performance degrades in case of heterogeneity.
Best criteria suffrage value algorithm	Chiang et al. [199]	Improve resource utilization. Reduced cost. Reduced Makespan.	Convergence is not considered. Response time is high.

Table 4: Summary of literature review based on traditional dynamic load balancing algorithms

Scheduling algorithms	Author(s) (Reference)	Advantages	Disadvantages
Dynamic algorithms	Tong et al. [85]	Real-time work distribution. Promote tolerance for errors. The system must only be in its current condition.	The nodes must be checked continuously. More complex.
Honey bee	LD et al. [200]	Improves throughput. Reduces response time.	Using a VM is essential for high priority tasks.
Ant-colony	Ragmani et al. [201]	Reduces the makespan. Jobs are autonomous. Complex in terms of computation.	Slow convergence.
Throttled load balancing	Duggal et al. [202]	Effective Collaboration of resources. Allocation of tasks involves a list.	Waiting time increases for tasks.
Genetic algorithm	Saadat et al. [188]	It provides a promising load balancing strategy and enables better utilization of resources.	The difficulty of computation has increased. Efficiency degrades when the search space is widened.

Table 5: Summary of metaheuristic based load balancing methods proposed in recent literature

Proposed method	Main objective	Area of interest	Addressed challenge(s)
Hybrid optimize algorithm (HOA) [191]	Effective distribution of VMs.	To provide optimal load balancing.	Targets cost, energy, response time, makespan, and execution time.
CMOA [190]	Improved load balancing.	To provide improved convergence with optimized QoS parameters.	Targets response time, makespan, and throughput.
LBAA [203]	Balancing the workload of the cloud.	To ensure high rate of resource utilization in Cloud.	Targets response time, makespan, and resource utilization.
Fuzzy logic-based GA [188]	Optimal load distribution in the cloud infrastructure.	Make appropriate selections for scheduling.	Facilitates the optimal scheduling of the request.
PSO algorithm [189]	Balancing the incoming traffic of cloud.	Minimizes response time.	Utilize the potential of numerous resources.
ACO algorithm with SVM [169]	Balancing the workload of the cloud with improved accuracy.	Reduces migration times, makespan, and response time.	Addresses stability and flexibility in cloud infrastructure.
ABC optimization [204]	Balancing the workload of the distributed systems.	Preemption of jobs to speed up responsiveness and operation.	Boosts performance parameters of quality.
GWO algorithm [205]	To achieve load balancing in cloud.	Resource distribution and load balancing.	Targets to reduce makespan as QoS parameter.
FPSO-GA algorithm [182]	Load balancing in cloud while saving energy.	Metaheuristic load balancing method for saving energy.	Reduces energy consumption, Execution Cost and Makespan.
Phasor PSO and dragonfly algorithm [206]	To produce the best possible resource allocation.	To investigate and exploit the search area effectively and efficiently.	More robust, reliable, and scalable algorithm.
Whale optimization improved [207]	Improved task scheduling in a cloud-based environment.	Manages resource distribution with reduced cost and makespan.	To increase the potential for globally optimized solution.

(Continued)

Table 5 (continued)

Proposed method	Main objective	Area of interest	Addressed challenge(s)
Balancer genetic algorithm (BGA) [208]	Task scheduling approach for cloud.	Optimizes the makespan and increases the throughput.	Improves makespan.
Improved GWO (iGWO) [152]	Load balancing in cloud computing.	Resource distribution with improved performance.	Improves convergences and reduces cost.

The identified load balancing techniques can also be compared in terms of their implementation, time complexity, and ability to adapt to changes in workload and associated challenges. The comparison of some of the most implemented techniques by researchers is presented in Table 6.

Table 6: Summary of Load Balancing methods on the basis of implementation, time complexity, adaptability and associated issue(s)

Algorithm	Implementation	Time complexity	Adaptability	Issue(s)
Static algorithm [84]	Easy	Low	Very limited	Current load and the performance of server is not considered.
Heuristic algorithm [91]	Average	Moderate/High	Limited	Run-time changes in server processing and the traffic is not considered.
Dynamic algorithms [55]	Complex	High	High	Scalability, response time are the associated challenges.
Metaheuristic algorithms [204]	Very complex	Very high	Very high	Scalability, computation overhead and convergence are the associated challenges.

There are certain limitations and constraints associated with the development and implementation of different load balancing algorithms. The algorithm's complexity, communication overhead, costs (setup cost, maintenance cost, communication cost and/or execution cost) [209] directly impact the implementation. In the case of implementing a global load-balancer, there are always scalability challenges, inefficient resource utilization, adaptability issues and the need for intelligent configurations. To effectively implement load balancing in a variety of cloud computing frameworks, addressing these difficulties calls for rigorous planning, monitoring, and adjustment [57].

When it comes to task scheduling, the time complexity of algorithms can vary significantly based on how they are implemented. Specifically, this depends on the categories of jobs they are balancing, the platforms they are working on, and the way load is identified and measured [210]. In general, the time complexity of static algorithms is lower than that of dynamic methods. This is because static algorithms fail to adjust in real-time to changes and distribute jobs based on predefined parameters. As a result their performance gradually deteriorates in scenarios where tasks are not evenly distributed [87]. On the other hand, heuristic-based and dynamic algorithms have a higher time complexity than static algorithms because they adjust mappings in real time. To maintain load balance, they continuously monitor the system and reassign duties as required. Due to their ability to adjust to changes, they perform better than static algorithms in dynamic environments. Additionally, Metaheuristics can have extremely high time complexity especially in their setup and iterative operations. The complexity primarily depends on the characteristics of the metaheuristics. However, this high time complexity comes with significantly improved performance as they identify near-optimal mappings in complicated and highly dynamic environments where standard methods fail and/or are ineffective [211]. The most effective algorithm will depend on the specific requirements and limitations of the environment in which it is implemented, it can be concluded. For instance, in an extremely dynamic and challenging environment, the improved performance and flexibility of metaheuristic-based algorithms totally justify their increased time complexities.

5 Comparative Analysis of the Reviewed Papers

This section addresses a detailed comparative analysis of the reviewed research in the context of load balancing in cloud computing (see Figs. 4–6). The QoS metrics that are typically used by authors when assessing load balancing methods are also compared. These features are essential when developing and building a load balancing strategy. The effectiveness of the algorithm in cloud-based services is evaluated using these metrics. These parameters play a major role in quantitively evaluating how optimally the proposed load balancing algorithm distributes the incoming traffic. Several of these metrics are used by researchers to evaluate the proposed approach; as a result, they should be modified to prevent load balancing problems in the particular context of cloud-based services [212]. Afzal et al. [213] pointed out the measures employed in the existing studies. The parameters of quality are presented below:

- Response time (RT) measures how quickly a device responds to a client request.
- Throughput (T) measures how fast customer queries are processed.
- Makespan (MS) measures how long it takes to complete the specified group of requests.
- Energy conservation (EC) shows the dependability and effectiveness of using electrical resources for various DCs operations, such as giving servers and cooling systems the power they demand.
- Scalability (S) measures keep up with the users' increasing and decreasing demands.
- Resource utilization (RU) measures the amount of resource consumption in the cloud DC.
- Cost (Ct) measures the expenses involved in distributing users' requests throughout the DCs.
- Convergence (C) refers to a point in the problem space where an objective function is optimized.

The comparative analysis of state-of-the-art algorithms based on QoS is presented in Table 7.

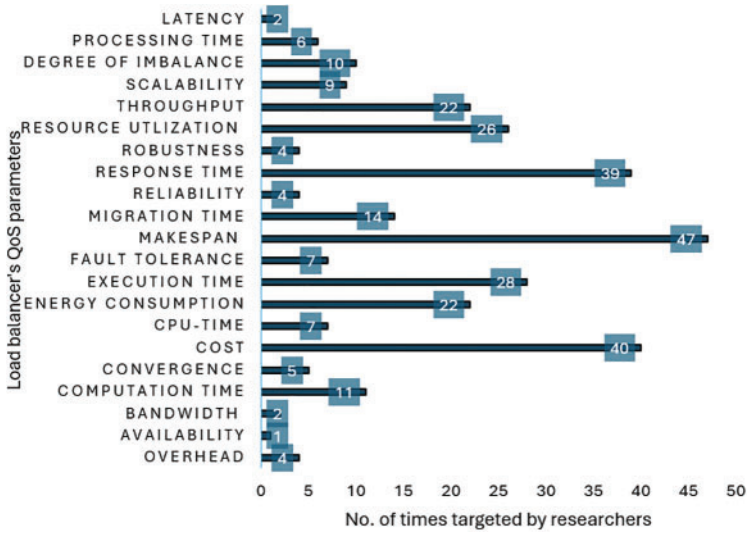

Figure 4: Load balancer's QoS parameters targeted by researchers from 2010–2024

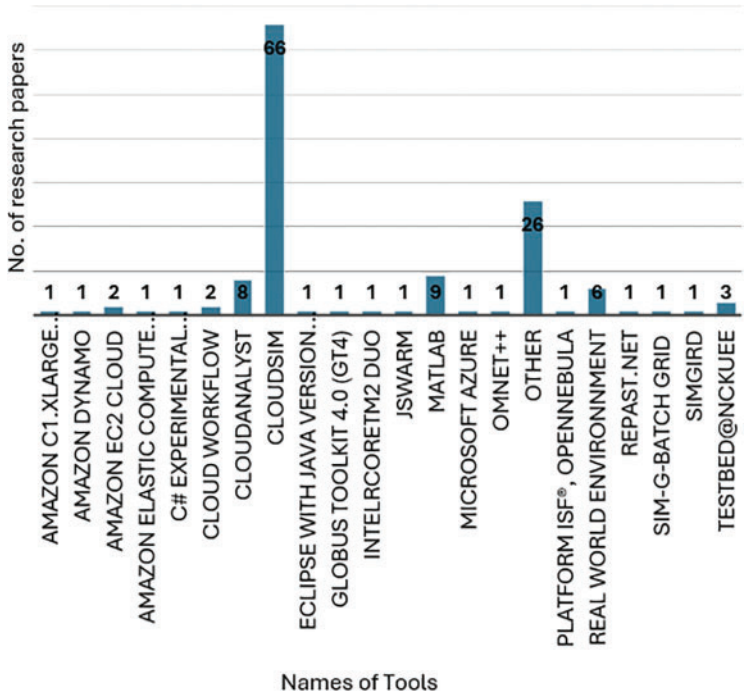

Figure 5: Trend of load balancing platforms from 2010–2024

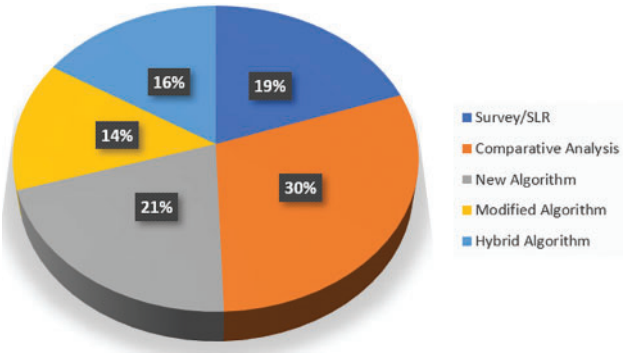

Figure 6: Percentage of the type of the targeted papers from 2010–2024

Table 7: Summary of targeted QoS parameters for load balancing By recent researchers

Algorithms (Year, reference)	RT	T	MS	EC	S	RU	C	C_t
Multi-Objective LB (2024, [194])	✓	✗	✓	✗	✗	✓	✗	✓
HOA (2024, [191])	✓	✗	✓	✓	✗	✓	✗	✓
CMOA(2023, [190])	✓	✓	✗	✗	✗	✗	✓	✓
LBAA (2023, [203])	✓	✗	✓	✗	✗	✓	✗	✗
PPSO-DA (2022, [178])	✓	✓	✗	✗	✓	✓	✗	✗
FPSO-GA (2022, [182])	✓	✓	✓	✗	✗	✓	✗	✓
Improved CSO (2022, [183])	✓	✓	✗	✗	✗	✓	✗	✗
BGA (2021, [208])	✓	✓	✗	✗	✗	✓	✗	✗
Fuzzy based GA (2020, [188])	✓	✓	✓	✗	✗	✓	✗	✓
Multi Agent GA(2020, [109])	✗	✓	✓	✗	✗	✓	✗	✗
Improved ABC (2020, [204])	✓	✗	✓	✗	✗	✓	✗	✓
Grey Wolf Algo. (2020, [214])	✓	✗	✓	✗	✗	✗	✓	✗
Improved GWO (2020, [215])	✓	✓	✓	✗	✗	✓	✓	✓
Improved Bat (2020, [216])	✓	✓	✗	✗	✗	✓	✗	✗
PSO (2019, [189])	✓	✗	✗	✗	✗	✓	✗	✗
IPSO (2019, [158])	✓	✗	✓	✓	✗	✓	✗	✓

From the literature review, a research gap in metaheuristic algorithms was identified based on the Quality of Service (QoS) parameters of Load balancers. Most researchers are focusing on response time, throughput, makespan, and resource utilization. However, there is still a need for further improvement as the global load continues to increase. The identified load balancing techniques enhance overall QoS by increasing availability, reliability, optimizing resource utilization, reducing latency, enabling scalability, ensuring fault tolerance, handling traffic, and providing dynamic adaptations for cloud-based applications and services. These load balancing techniques can be integrated with existing cloud management platforms and tools. However, the ease and specificity of integration may vary depending on various criteria, including compatibility with the cloud infrastructure, modification

of parameters, configuration of network settings, adherence to load balancing procedure rules, utilization of automation tools, ensuring security compliance, referring to documentation, and conducting tests on various scenarios for better results.

Most algorithms noticeably lack energy consumption components. An essential gap in the current era of energy efficiency, especially for distributed and mobile computing systems, is the lack of algorithms that prioritize low energy consumption (EC). This is particularly important for applications in settings with constrained energy supplies or high energy conservation priorities. Scalability is another aspect of the algorithms that is underrepresented. The ability to scale efficiently is essential as data quantities and computational needs continue to increase. There is a need to create algorithms that can effectively manage vast amounts of data or adapt to different workload sizes, as most of the listed methods do not specifically target scalability. Additionally, all the recently published work considers response time (RT), makespan (MS), resource utilization (RU) and cost (C_t) as essential QoS parameters that need improvement. Although several algorithms exhibit real-time processing capabilities (RT), they often neglect to consider EC and scalability (S) at the same time. There is a lot of research potential in creating algorithms that are scalable, energy-efficient, and capable of delivering real-time performance. It seems that there are not many algorithms that can successfully satisfy multiple requirements simultaneously. For instance, Table 7 does not clearly provide an approach that is scalable, energy-efficient, and addresses both makespan (MS) and throughput (T). This indicates a need for more comprehensive algorithms that strike a balance between several important performance indicators. Therefore, there is still a need for the identified load balancing techniques to be extended or modified to address multiple quality of service parameters at the same time. These extended, modified or hybrid load balancing techniques will then offer multi-objective algorithms that can more effectively tackle the emerging challenges or trends in cloud computing.

Fig. 4 illustrates that, of the load balancer's enlisted QoS parameters, makespan is considered the most, i.e., 16% of the time, followed by cost (14% of the time), response time (13% of the time), execution time (9% of the time), resource utilization (8.5% of the time), throughput (approximate 7% of the time), and energy conservation (about 7% of the time). Only about 4.5% of the time is considered migration time. Degree of Imbalance and processing time are considered around 3% of the time, respectively, and computation time is considered 3.8% of the time. About 2% of the time, CPU-time, fault tolerance, and scalability are taken into consideration. The least targeted performance parameters are overhead, availability, bandwidth, convergence, reliability, and robustness. Fig. 5 presents the selected platforms by researchers to evaluate the performance of the proposed algorithm. It concludes that most of the load balancing research work is performed by using simulation tools and among the simulation tools CloudSim is the most preferred simulation tool. Additionally, only 11 papers deploy the proposed algorithms in a real-world environment (among which 5 cases use Amazon services). Fig. 6 shows the percentage of the type (i.e., review, Comparative Analysis, Novel Algorithm, Improved Algorithm or Hybrid Approach) of the targeted papers from 2010–2024. Fig. 7 shows the performance analysis of the Static Algorithm (SA), Traditional Dynamic Algorithm (TDA) and Metaheuristic based Dynamic Algorithm (MDA) based on response time. The simulation is conducted on CloudSim tool by using eclipse IDE with Cloudlets (i.e., user requests) are 50,100,300 and 500, respectively. The configurational characteristics of DCs, VMs, and cloudlets are compiled in Table 8. In this simulation R-R algorithm is chosen as SA, WLC algorithm is chosen as TDA and ACO algorithm is chosen as MDA.

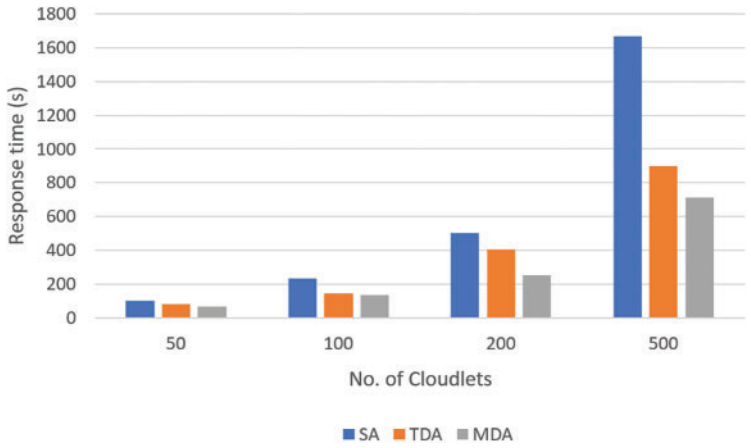

Figure 7: Response times of algorithms (When DCs = 3 and VMs = 25)

Table 8: The characteristics of a DC, VMs, and cloudlets

Components	Characteristics	Value(s)
DCs	Architecture	x86
	Operating system	Linux
	VMM	Xen
	Cost of processing	3.0
	Cost of memory	0.05
VMs	Image size (MB)	10000
	Ram (MB)	512
	Processing speed (MIPs)	1000
	Bandwidth	1000
	Number of CPUs	1
	VMM name	Xen
Cloudlets	Total length	1000
	File size	300
	Output size	300
	Required processing elements to process	1
	Total number of allocated tasks	10

Fig. 7 shows that SA and TDA perform better under low workloads, but MDA still outperforms them. As workloads increase MDA exhibits improved response time to a considerable degree, suggesting its suitability for environments with dynamic and/or high user retention. This indicates that MDA is an effective load balancing approach to maintain reduced response times and, consequently, improved quality of service for cloud environments expecting unpredictable or increasing customer demand.

6 Conclusions and Recommendations

Cloud environment services are practically dynamic and unconstrained in operation. Load balancing acts as a crucial factor concurrent to makespan for effective task scheduling, highlighting the importance of multi-objective optimization. Traditional static scheduling algorithms often fall short in meeting the realistic resource scheduling criteria of cloud environments, leading to significant resource wastage [208]. As a result, the growing need for dynamic approaches encourages the development of metaheuristic-based algorithms. These algorithms represent a major area of research, focusing on improving the effectiveness of state-of-the-art resource scheduling methods for load balancing. This study extensively explores the background literature concerning load balancing, with particular attention given to metaheuristic-based algorithms. To explore a large search space of multiple optimal alternatives, meta-heuristic algorithms are the most preferred option. Still, they come with several associated challenges [217,218], such as difficulty in tracking dynamic allocation, increased computational time, high computational costs, occasional difficulty in finding a global optimum, susceptibility to being stuck in local optima, slow convergence speed, and limited accuracy. Hence, various hybrid metaheuristic-based load balancing algorithms have been proposed to leverage their complementary strengths. However, none have yet achieved improved convergence with scalability. This review identifies a critical research gap that necessitates addressing the challenge of load balancing in cloud computing, aiming for both global optimization and rapid convergence simultaneously.

The systematic review presented herein highlights several key recommendations that both practitioners and researchers should consider for the effective implementation of load balancing techniques in cloud computing. These recommendations encompass various aspects, including cost considerations, types of applications, scalability needs, Quality of Service (QoS) parameters, fault tolerance, associated overheads, adaptability, and security requirements. Furthermore, it emphasizes the importance of staying abreast of the latest advancements and keeping comprehensive records. Researchers are advised to first gain a thorough understanding of the set-up, objectives, and associated costs, aiming to achieve an optimal balance between cost and functionality. Security considerations are also paramount across all environments. Moreover, it is advisable to conduct compatibility assessments among platforms, tools, and services when selecting a load balancing technique for any given infrastructure. It also emphasizes how crucial it is to choose the best load balancing technique strategically, considering the requirements of cloud-based apps and their deployment circumstances. To ensure that user satisfaction achieves QoS specifications, practitioners must commence their efforts by accurately defining the primary goals of the cloud application. Prior to implementation, a thorough analysis of the deployment environment (i.e., public, private, or hybrid) is also necessary. This entails knowing each type's inherent qualities as well as how they fit the needs of the application and security regulations. An important part of this selection practice is also figuring out the typical pattern of incoming user requests. Practitioners can select a load balancing method that guarantees efficient utilization of resources and maintains high levels of service uptime and responsiveness by examining these patterns, which helps them better predict variations in load. Consequently, with a thorough investigation, it is advised that future studies focus on the creation of dynamic, adaptive load balancing strategies that are capable of sensibly adapting to shifting workload dynamics and deployment conditions. These developments have the potential to substantially enhance cloud-based services' scalability, effectiveness, and overall performance. Furthermore, load balancing solutions need to be reviewed and improved upon to maintain the rapid evolution of the cloud computing landscape and the emergence of new technologies and architectures. By consistently introducing new ideas and adapting to changing circumstances, CSPs will attempt to satisfy user expectations and expand the realm of possibilities that are feasible in cloud computing.

Several core improvements are required to improve the performance of load balancing algorithms for cloud-based environments in the coming years. Practitioners and/or researchers must accept the fact that with the increase in dynamic incoming traffic, it is highly necessary to develop enhanced algorithms for efficient resource management and optimal load distribution. The designed algorithms must be able to identify the optimum tracking frequency ranges, variable thresholds, low migration cost, and data transmission overheads. Furthermore, there are computational overheads related to various essential operations, including VM migration, task migration, and system state monitoring. All load balancing activities need to be optimized to facilitate the improved scheduling of resources. To anticipate potential unbalanced conditions with high accuracy, more efficient workload estimation techniques must be devolved. Considering the literature, it can be inferred that, even though network capacity is crucial to cloud computing due to significant network traffic, the necessary consideration of networking elements' effective utilization has not been made. Significant implications like network outages, loss of information, and delays in communication may result from this. Therefore, more accurate and improved load balancing algorithms should be created to effectively utilize the network's resources. Most load balancing methods for cloud computing are currently created and validated using simulation toolkits.

The review can be improved in the future by providing a practical implementation of algorithms in real-world scenarios. To determine the applicability of any load balancing algorithm in an actual cloud configuration, it must undergo real-world execution. In order to accurately compare proposed load balancing strategies, the effectiveness of the techniques for load balancing needs to be assessed in relation to the standard setup. Inter-cloud load balancing solutions, in which many cloud service providers may cooperate, are necessary to address the constantly increasing and highly unpredictable service needs in the cloud.

Acknowledgement: I thank my supervisor and co-supervisor for their guidance and support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: selection of load balancing issue in cloud computing and study conception, design, and detailed literature review: Darakhshan Syed; identification of research gaps: Darakhshan Syed, Ghulam Muhammad, Safdar Rizvi; analysis of the reviewed papers: Darakhshan Syed and Safdar Rizvi; final draft manuscript preparation: Darakhshan Syed. All authors reviewed the content and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog," *IEEE Access*, vol. 7, pp. 150936–150948, 2019. doi: 10.1109/ACCESS.2019.2947652.

[2] T. DeStefano, R. Kneller, and J. Timmis, "Cloud computing and firm growth," *Rev. Econ. Stat.*, 2020. doi: 10.1162/rest_a_01393.

[3] Y. Chen, X. Li, and F. Chen, "Overview and analysis of cloud computing research and application," in *Int. Conf. E-Business and E-Gov. (ICEE)*, IEEE, 2011. doi: 10.1109/ICEBEG.2011.5881819.
[4] D. Syed, U. B. Masood, U. E. Farwa, and M. Khurram, "Cloud based smart irrigation for agricultural area of Pakistan," *Comput. Eng. Appl. J.*, vol. 4, no. 3, pp. 153–164, 2015. doi: 10.18495/COMENGAPP.V4I3.152.
[5] B. Langmead and A. Nellore, "Cloud computing for genomic data analysis and collaboration," *Nat. Rev. Genet.*, vol. 19, no. 4, pp. 208–219, 2018. doi: 10.1038/nrg.2017.113.
[6] A. T. Velte, T. J. Velte, and R. C. Elsenpeter, *Cloud Computing, A Practical Approach*. New York: McGraw-Hill Inc, 2009.
[7] F. Ebadifard and S. M. Babamir, "A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment," *Concurr. Comput.*, vol. 30, no. 12, pp. e4368, 2018. doi: 10.1002/cpe.4368.
[8] M. S. Al Reshan *et al.*, "A fast converging and globally optimized approach for load balancing in cloud computing," *IEEE Access*, vol. 11, pp. 11390–11404, 2023. doi: 10.1109/ACCESS.2023.3241279.
[9] D. Syed, N. Islam, M. H. Shabbir, and S. B. Manzar, "Applications of big data in smart health systems," in *Handbook of Research on Mathematical Modeling for Smart Healthcare Systems*, Hershey, Pennsylvania, USA: IGI Global, 2022, pp. 52–85. doi: 10.4018/978-1-6684-4580-8.ch004.
[10] A. Jangra and N. Mangla, "An efficient load balancing framework for deploying resource schedulingin cloud based communication in healthcare," *Meas. Sens.*, vol. 25, pp. 100584, 2023. doi: 10.1016/j.measen.2022.100584.
[11] P. Yang, L. Zhang, H. Liu, and G. Li, "Reducing idleness in financial cloud services via multi-objective evolutionary reinforcement learning based load balancer," *Sci. China Inf. Sci.*, vol. 67, no. 2, pp. 1–21, 2024. doi: 10.1007/s11432-023-3895-3.
[12] V. Talukdar *et al.*, "Load balancing techniques in cloud computing," in *Emerging Trends in Cloud Computing Analytics, Scalability, and Service Models*, Hershey, Pennsylvania, USA, IGI Global, 2024, pp. 105–134. doi: 10.4018/979-8-3693-0900-1.
[13] P. A. Malla, S. Sheikh, M. Shahid, and S. U. Mushtaq, "Energy-efficient sender-initiated threshold-based load balancing (e-STLB) in cloud computing environment," *Concurr. Comput. Pract. Exp.*, vol. 36, no. 5, pp. e7943, 2024. doi: 10.1002/cpe.7943.
[14] S. Kaur and A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment," *Int. J. Inf. Technol. Comput. Sci. (IJITCS)*, vol. 4, no. 10, pp. 74–79, 2012. doi: 10.5815/ijitcs.2012.10.09.
[15] D. Agarwal and S. Jain, "Efficient optimal algorithm of task scheduling in cloud computing environment," 2014. doi: 10.48550/arXiv.1404.2076.
[16] A. Hussain, M. Aleem, M. A. Iqbal, and M. A. Islam, "SLA-RALBA: Cost-efficient and resource-aware load balancing algorithm for cloud computing," *J. Supercomput.*, vol. 75, no. 10, pp. 6777–6803, 2019. doi: 10.1007/s11227-019-02916-4.
[17] T. Deepa and D. Cheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing," in *Int. Conf. Energy, Commun., Data Anal. Soft Comput. (ICECDS)*, IEEE, 2017. doi: 10.1109/ICECDS.2017.8390086.
[18] Z. Zhou, F. Li, H. Zhu, H. Xie, J. H. Abawajy and M. U. Chowdhury, "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," *Neural Comput. Appl.*, vol. 32, pp. 1531–1541, 2020. doi: 10.1007/s00521-019-04119-7.
[19] R. Kaur, "A review of computing technologies: Distributed, utility, cluster, grid and cloud computing," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 2, pp. 144–148, 2015.
[20] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. Oxford, England. John Wiley & Sons, Inc., Blackwell Scientific Publishing, 1993.
[21] I. A. Saroit and D. Tarek, "LBCC-Hung: A load balancing protocol for cloud computing based on Hungarian method," *Egypt. Inform. J.*, vol. 24, no. 3, pp. 100387, 2023. doi: 10.1016/j.eij.2023.100387.

[22] M. A. Arfeen, K. Pawlikowski, and A. Willig, "A framework for resource allocation strategies in cloud computing environment," in *IEEE 35th Annu. Comput. Softw. Appl. Conf. Workshops*, IEEE, 2011. doi: 10.1109/COMPSACW.2011.52.

[23] X. Sun, N. Ansari, and R. Wang, "Optimizing resource utilization of a data center," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2822–2846, 2016. doi: 10.1109/COMST.2016.2558203.

[24] S. Singh and I. Chana, "Cloud resource provisioning: Survey, status and future research directions," *Knowl. Inf. Syst.*, vol. 49, pp. 1005–1069, 2016. doi: 10.1007/s10115-016-0922-3.

[25] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, 2010. doi: 10.1109/JPROC.2010.2060451.

[26] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par Parallel Process. 19th Int. Conf.*, Aachen, Germany, Springer, Aug. 26–30, 2013. doi: 10.1007/978-3-642-40047-6_33.

[27] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012. doi: 10.1016/j.future.2011.04.017.

[28] T. Kaur and I. Chana, "Energy efficiency techniques in cloud computing: A survey and taxonomy," *ACM Comput. Surv. (CSUR)*, vol. 48, no. 2, pp. 1–46, 2015. doi: 10.1145/2742488.

[29] S. Sagar, A. Choudhary, M. S. A. Ansari, and M. C. Govil, "A survey of energy-aware server consolidation in cloud computing," in *Evol. Comput. Intell.: Proc. 10th Int. Conf. Front. Intell. Comput.: Theory and Appl. (FICTA 2022)*, Springer, 2023, vol. 326, pp. 381–391. doi: 10.1007/978-981-19-7513-4_34.

[30] R. Gogulan, A. Kavitha, and U. K. Kumar, "An multiple pheromone algorithm for cloud scheduling with various QOS requirements," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 9, no. 3, pp. 232, 2012.

[31] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in *2012 Second Symp. Network Cloud Comput. Appl.*, IEEE, 2012. doi: 10.1109/NCCA.2012.29.

[32] N. J. Kansal and I. Chana, "Existing load balancing techniques in cloud computing: A systematic review," *J. Inf. Syst. Commun.*, vol. 3, no. 1, pp. 87, 2012.

[33] N. J. Kansal and I. Chana, "Cloud load balancing techniques: A step towards green computing," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 9, no. 1, pp. 238–246, 2012.

[34] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," *J. Netw. Comput. Appl.*, vol. 71, pp. 86–98, 2016. doi: 10.1016/j.jnca.2016.06.003.

[35] T. Diaby and B. B. Rad, "Cloud computing: A review of the concepts and deployment models," *Int. J. Inf. Technol. Comput. Sci.*, vol. 9, no. 6, pp. 50–58, 2017. doi: 10.5815/ijitcs.2017.06.07.

[36] M. A. Elmagzoub, D. Syed, A. Shaikh, N. Islam, A. Alghamdi and S. Rizwan, "A survey of swarm intelligence based load balancing techniques in cloud computing environment," *Electronics*, vol. 10, no. 21, pp. 2718, 2021. doi: 10.3390/electronics10212718.

[37] M. Paul and G. Sanyal, "Survey and analysis of optimal scheduling strategies in cloud environment," in *2011 World Congr. Inf. Commun. Technol.*, IEEE, 2011. doi: 10.1109/WICT.2011.6141347.

[38] C. W. Tsai and J. J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," *IEEE Syst. J.*, vol. 8, no. 1, pp. 279–291, 2013. doi: 10.1109/JSYST.2013.2256731.

[39] D. Kashyap and J. Viradiya, "A survey of various load balancing algorithms in cloud computing," *Int. J. Sci. Technol. Res.*, vol. 3, no. 11, pp. 115–119, 2014.

[40] S. B. Shaw and A. Singh, "A survey on scheduling and load balancing techniques in cloud computing environment," in *2014 Int. Conf. Comput. Commun. Technol. (ICCCT)*, IEEE, 2014. doi: 10.1109/ICCCT.2014.7001474.

[41] A. Hamidi and R. Astya, "Load balancing in cloud computing using meta-heuristic algorithm: A review," in *2022 9th Int. Conf. Comput. Sustain. Glob. Dev. (INDIACom)*, IEEE, 2022. doi: 10.23919/INDIACom54597.2022.9763131.

[42] S. Ebneyousef and A. Shirmarz, "A taxonomy of load balancing algorithms and approaches in fog computing: A survey," *Clust. Comput.*, pp. 1–22, 2023. doi: 10.1007/s10586-023-03982-3.
[43] A. Asghari and M. K. Sohrabi, "Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet," *Comput. Sci. Rev.*, vol. 51, pp. 100616, 2024. doi: 10.1016/j.cosrev.2023.100616.
[44] S. I. Abbas and P. Verma, "Cloud computing load balancing technology: Mapping study," in *Artificial Intelligence, Blockchain, Computing and Security*. Howick Place, London: CRC Press, 2024, vol. 2, pp. 180–185.
[45] Z. S. Ageed and S. R. Zeebaree, "Distributed systems meet cloud computing: A review of convergence and integration," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 11s, pp. 469–490, 2024.
[46] A. Pradhan, S. K. Bisoy, and A. Das, "A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 34, no. 8, pp. 4888–4901, 2022. doi: 10.1016/j.jksuci.2021.01.003.
[47] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl. Based Syst.*, vol. 159, pp. 20–50, 2018. doi: 10.1016/j.knosys.2018.06.001.
[48] S. A. A. Naqvi, N. Javaid, H. Butt, M. B. Kamal, A. Hamza and M. Kashif, "Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid," in *Adv. Netw.-Based Inf. Syst.: The 21st Int. Conf. Netw.-Based Inf. Syst. (NBiS-2018)*, Springer, 2019, vol. 22, pp. 700–711. doi: 10.1007/978-3-319-98530-5_61.
[49] A. Javadpour *et al.*, "An energy-optimized embedded load balancing using DVFS computing in cloud data centers," *Comput. Commun.*, vol. 197, pp. 255–266, 2023. doi: 10.1016/j.comcom.2022.10.019.
[50] H. Alshahrani *et al.*, "Sustainability in blockchain: A systematic literature review on scalability and power consumption issues," *Energies*, vol. 16, no. 3, pp. 1510, 2023. doi: 10.3390/en16031510.
[51] D. Jian, "Cloud model and ant colony optimization based QoS routing algorithm for wireless sensor networks," in *Adv. Technol. Teach.-Proc. 2009 3rd Int. Conf. Teach. Comput. Sci. (WTCS 2009), Intell. Ubiquitous Comput. Educ.*, Springer, 2012, vol. 116. doi: 10.1007/978-3-642-11276-8_23.
[52] S. T. Waghmode and B. M. Patil, "Adaptive load balancing in cloud computing environment," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1s, pp. 209–217, 2023.
[53] C. C. Lin, H. H. Chin, and D. J. Deng, "Dynamic multiservice load balancing in cloud-based multimedia system," *IEEE Syst. J.*, vol. 8, no. 1, pp. 225–234, 2013. doi: 10.1109/JSYST.2013.2256320.
[54] Y. Sahu and R. Pateriya, "Cloud computing overview with load balancing techniques," *Int. J. Comput. Appl.*, vol. 65, no. 24, pp. 40–44, 2013.
[55] V. R. Kanakala, V. K. Reddy, and K. Karthik, "Performance analysis of load balancing techniques in cloud computing environment," in *2015 IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, IEEE, 2015. doi: 10.1109/ICECCT.2015.7226052.
[56] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, pp. 106040, 2019. doi: 10.1016/j.cie.2019.106040.
[57] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," in *Comput. Intell. Multimed. Big Data Cloud with Eng. Appl.*, Academic Press, 2018, pp. 185–231. doi: 10.1016/B978-0-12-813314-9.00010-4.
[58] S. T. Milan, L. Rajabion, H. Ranjbar, and N. J. Navimipour, "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments," *Comput. Oper. Res.*, vol. 110, pp. 159–187, 2019. doi: 10.1016/j.cor.2019.05.022.
[59] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. UK: Luniver Press, 2010.
[60] K. V. Kumar and A. Rajesh, "Multi-objective load balancing in cloud computing: A meta-heuristic approach," *Cybern. Syst.*, vol. 54, no. 8, pp. 1466–1493, 2023. doi: 10.1080/01969722.2022.2145656.
[61] H. Rahmawan and Y. S. Gondokaryono, "The simulation of static load balancing algorithms," in *2009 Int. Conf. Electr. Eng. Inf.*, Bangi, Malaysia, IEEE, 2009. doi: 10.1109/ICEEI.2009.5254739.
[62] R. Gao and J. Wu, "Dynamic load balancing strategy for cloud computing with ant colony optimization," *Future Internet*, vol. 7, no. 4, pp. 465–483, 2015. doi: 10.3390/fi7040465.

[63] S. Chhabra and A. K. Singh, "Dynamic resource allocation method for load balance scheduling over cloud data center networks," *J. Web Eng.*, vol. 20, no. 8, pp. 2269–2284, 2021. doi: 10.13052/jwe1540-9589.2083.
[64] H. B. Alla, S. B. Alla, A. Touhafi, and A. Ezzati, "A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment," *Cluster Comput.*, vol. 21, no. 4, pp. 1797–1820, 2018. doi: 10.1007/s10586-018-2811-x.
[65] M. N. Aktan and H. Bulut, "Metaheuristic task scheduling algorithms for cloud computing environments," *Concurr. Comput.: Pract. and Exp.*, vol. 34, no. 9, pp. e6513, 2022. doi: 10.1002/cpe.6513.
[66] L. Heilig, E. Lalla-Ruiz, S. Voß, and R. Buyya, "Metaheuristics in cloud computing," *Softw. Pract. Exp.*, vol. 48, pp. 1729–1733, 2018. doi: 10.1002/spe.2628.
[67] Y. Wen, H. Xu, and J. Yang, "A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system," *Inf. Sci.*, vol. 181, no. 3, pp. 567–581, 2011. doi: 10.1016/j.ins.2010.10.001.
[68] V. Sesum-Cavic and E. Kühn, "Applying swarm intelligence algorithms for dynamic load balancing to a cloud based call center," in *2010 Fourth IEEE Int. Conf. Self-Adapt. Self-Organ. Syst.*, Budapest, Hungary, IEEE, 2010. doi: 10.1109/SASO.2010.19.
[69] S. H. Li and J. I. G. Hwang, "Bidirectional ant colony optimization algorithm for cloud load balancing," in *Proc. 2nd Int. Conf. Intell. Technol. Eng. Syst. (ICITES2013)*, Springer, 2014, vol. 293, pp. 907–913. doi: 10.1007/978-3-319-04573-3_111.
[70] H. R. Faragardi, S. Dehnavi, T. Nolte, M. Kargahi, and T. Fahringer, "An energy-aware resource provisioning scheme for real-time applications in a cloud data center," *Softw. Pract. Exp.*, vol. 48, no. 10, pp. 1734–1757, 2018. doi: 10.1002/spe.2592.
[71] A. Yousefipour, A. M. Rahmani, and M. Jahanshahi, "Energy and cost-aware virtual machine consolidation in cloud computing," *Softw. Pract. Exp.* vol. 48, no. 10, pp. 1758–1774, 2018. doi: 10.1002/spe.2585.
[72] K. Li, "Optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds," *Softw. Pract. Exp.*, vol. 48, no. 10, pp. 1805–1819, 2018. doi: 10.1002/spe.2584.
[73] A. G. Delavar and Y. Aryan, "HSGA: A hybrid heuristic algorithm for workflow scheduling in cloud systems," *Cluster Comput.*, vol. 17, pp. 129–137, 2014. doi: 10.1007/s10586-013-0275-6.
[74] S. Lata and D. Singh, "A hybrid approach for cloud load balancing," in *2022 2nd Int. Conf. Adv. Comput. Innov. Technol. Eng. (ICACITE)*, Greater Noida, India, IEEE, 2022. doi: 10.1109/ICACITE53722.2022.9823569.
[75] M. Sumathi, N. Vijayaraj, S. P. Raja, and M. Rajkamal, "HHO-ACO hybridized load balancing technique in cloud computing," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1357–1365, 2023. doi: 10.1007/s41870-023-01159-0.
[76] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Metaheuristic approaches to virtual machine placement in cloud computing: A review," in *2016 15th Int. Symp. Parallel and Distrib. Comput. (ISPDC)*, Fuzhou, China, IEEE, 2016. doi: 10.1109/ISPDC.2016.37.
[77] M. Rahman, X. Li, and H. Palit, "Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment," in *2011 IEEE Int. Symp. Parallel Distrib. Process. Workshops and Phd Forum*, Anchorage, AK, USA, IEEE, 2011. doi: 10.1109/IPDPS.2011.243.
[78] S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu and A. Abraham, "Hybrid job scheduling algorithm for cloud computing environment," in *Proc. Fifth Int. Conf. Innov. Bio-Inspired Comput. Appl.*, Cham, Springer, 2014, vol. 303, pp. 43–52. doi: 10.1007/978-3-319-08156-4_5.
[79] A. Javadpour, G. Wang, and S. Rezaei, "Resource management in a peer to peer cloud network for IoT," *Wirel. Pers. Commun.*, vol. 115, no. 3, pp. 2471–2488, 2020. doi: 10.1007/s11277-020-07691-7.
[80] K. K. Sowjanya and S. Mouleeswaran, "Load balancing algorithms in cloud computing," in *Proc. Int. Conf. Cogn. Intell. Comput.*, Singapore, Springer, 2023, vol. 2, pp. 483–493. doi: 10.1007/978-981-19-2358-6_45.
[81] A. Hota, S. Mohapatra, and S. Mohanty, "Survey of different load balancing approach-based algorithms in cloud computing: A comprehensive review," in *Comput. Intell. Data Mining: Proc. Int. Conf. CIDM 2017*, Singapore, 2019, vol. 722, pp. 99–110. doi: 10.1007/978-981-10-8055-5_10.

[82] P. B. Soundarabai, R. K. Sahai, J. Thriveni, K. R. Venugopal, and L. M. Patnaik, "Comparative study on load balancing techniques in distributed systems," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 6, no. 1, pp. 53–60, 2012.

[83] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," in *2013 Nat. Conf. Parallel Comput. Technol. (PARCOMPTECH)*, Bangalore, India, IEEE, 2013. doi: 10.1109/ParCompTech.2013.6621389.

[84] T. Desai and J. Prajapati, "A survey of various load balancing techniques and challenges in cloud computing," *Int. J. Sci. Technol. Res.*, vol. 2, no. 11, pp. 158–161, 2013.

[85] R. Tong and X. Zhu, "A load balancing strategy based on the combination of static and dynamic," in *2010 2nd Int. Workshop Database Technol. Appl.*, Wuhan, China, IEEE, 2010. doi: 10.1109/DBTA.2010.5658951.

[86] S. C. Wang, K. Q. Yan, W. P. Liao, and S. S. Wang, "Towards a load balancing in a three-level cloud computing network," in *2010 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Chengdu, China, IEEE, 2010. doi: 10.1109/ICCSIT.2010.5563889.

[87] S. Aslam and M. A. Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques," in *2015 Nat. Softw. Eng. Conf. (NSEC)*, Rawalpindi, Pakistan, IEEE, 2015. doi: 10.1109/NSEC.2015.7396341.

[88] F. Ramezani, J. Lu, and F. K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," *Int. J. Parallel Program.*, vol. 42, pp. 739–754, 2014. doi: 10.1007/s10766-013-0275-4.

[89] S. Kapoor and C. Dabas, "Cluster based load balancing in cloud computing," in *2015 Eighth Int. Conf. Contemp. Comput. (IC3)*, Noida, India, IEEE, 2015. doi: 10.1109/IC3.2015.7346656.

[90] A. Jaiswal and S. Jain, "An approach towards the dynamic load management techniques in cloud computing environment," in *2014 Int. Conf. Power, Autom. Commun. (INPAC)*, Amravati, India, IEEE, 2014. doi: 10.1109/INPAC.2014.6981147.

[91] S. G. Domanal and G. R. M. Reddy, "Load balancing in cloud computingusing modified throttled algorithm," in *2013 IEEE Int. Conf. Cloud Comput. Emerging Markets (CCEM)*, Bangalore, India, IEEE, 2013. doi: 10.1109/CCEM.2013.6684434.

[92] H. Singh, S. Tyagi, P. Kumar, S. S. Gill, and R. Buyya, "Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions," *Simul. Model. Pract. Theory*, vol. 111, pp. 102353, 2021. doi: 10.1016/j.simpat.2021.102353.

[93] D. Syed *et al.*, "A comparative analysis of metaheuristic techniques for high availability systems (September 2023)," *IEEE Access*, vol. 12, pp. 7382–7398, 2024. doi: 10.1109/ACCESS.2024.3352078.

[94] P. S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in *2010 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Perth, WA, Australia, IEEE, 2010. doi: 10.1109/AINA.2010.31.

[95] Z. Wu, Z. Ni, L. Gu, and X. Liu, "A revised discrete particle swarm optimization for cloud workflow scheduling," in *2010 Int. Conf. Comput. Intell. Secur.*, Nanning, China, IEEE, 2010. doi: 10.1109/CIS.2010.46.

[96] F. Zamfirache, D. Zaharie, and C. Craciun, "Nature inspired metaheuristics for task scheduling in static and dynamic computing environments," *Nature*, vol. 55, no. 69, pp. 133–142, 2010.

[97] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh and R. Rastogi, "Load balancing of nodes in cloud using ant colony optimization," in *2012 UKSim 14th Int. Conf. Comput. Model. Simul.*, Cambridge, UK, IEEE, 2012. doi: 10.1109/UKSim.2012.11.

[98] M. J. Csorba, H. Meling, and P. E. Heegaard, "Ant system for service deployment in private and public clouds," in *Proc. 2nd Workshop on Bio-Inspired Algorithms for Distrib. Syst.*, Washington DC, USA, 2010, pp. 19–28. doi: 10.1145/1809018.1809024.

[99] V. Sesum-Cavic and E. Kühn, "Comparing configurable parameters of swarm intelligence algorithms for dynamic load balancing," in *2010 Fourth IEEE Int. Conf. Self-Adapt. Self-Organ. Syst. Workshop*, Budapest, Hungary, IEEE, 2010. doi: 10.1109/SASOW.2010.12.

[100] C. Szabo and T. Kroeger, "Evolving multi-objective strategies for task allocation of scientific workflows on public clouds," in *IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, IEEE, 2012. doi: 10.1109/CEC.2012.6256556.
[101] J. Kołodziej and S. U. Khan, "Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment," *Inf. Sci.*, vol. 214, pp. 1–19, 2012. doi: 10.1016/j.ins.2012.05.016.
[102] Z. Zhang and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation," in *2010 the 2nd Int. Conf. Ind. Mechatron. Autom.*, Wuhan, China, IEEE, 2010. doi: 10.1109/ICINDMA.2010.5538385.
[103] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *2010 3rd Int. Symp. Parallel Archit., Algorithms And Program.*, Liaoning, China, IEEE, 2010. doi: 10.1109/PAAP.2010.65.
[104] Y. Ge and G. Wei, "GA-based task scheduler for the cloud computing systems," in *2010 Int. Conf. Web Inf. Syst. Min.*, Sanya, China, IEEE, 2010. doi: 10.1109/WISM.2010.87.
[105] K. Mukherjee and G. Sahoo, "Performance analysis of cloud computing using multistage ant system," *Int. J. Comput. Appl.*, vol. 1, no. 20, pp. 70–74, 2010.
[106] R. M. Chen and C. M. Wang, "Project scheduling heuristics-based standard PSO for task-resource assignment in heterogeneous grid," in *Abstract and Applied Analysis*, Cairo, Egypt, Hindawi, 2011, vol. 2011. doi: 10.1155/2011/589862.
[107] M. E. Frincu and C. Craciun, "Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments," in *2011 Fourth IEEE Int. Conf. Util. Cloud Comput.*, Melbourne, VIC, Australia, IEEE, 2011. doi: 10.1109/UCC.2011.43.
[108] G. W. You, S. W. Hwang, and N. Jain, "Scalable load balancing in cluster storage systems," in *ACM/IFIP/USENIX Int. Conf. Distrib. Syst. Platforms and Open Distrib. Process.*, Berlin, Heidelberg, Springer, 2011, vol. 7049, doi: 10.1007/978-3-642-25821-3_6.
[109] K. Zhu, H. Song, L. Liu, J. Gao, and G. Cheng, "Hybrid genetic algorithm for cloud computing applications," in *2011 IEEE Asia-Pac. Serv. Comput. Conf.*, Jeju, Korea (South), IEEE, 2011. doi: 10.1109/APSCC.2011.66.
[110] Y. Xu, L. Wu, L. Guo, Z. Chen, L. Yang and Z. Shi, "An intelligent load balancing algorithm towards efficient cloud computing," in *Proc. AI Data Cent. Manag. Cloud Comput.*, 2008, pp. 27–32.
[111] M. Paul, D. Samanta, and G. Sanyal, "Dynamic job scheduling in cloud computing based on horizontal load balancing," *Int. J. Comput. Technol. Appl. (IJCTA)*, vol. 2, no. 5, pp. 1552–1556, 2011.
[112] Y. H. Zhang, L. Feng, and Z. Yang, "Optimization of cloud database route scheduling based on combination of genetic algorithm and ant colony algorithm," *Procedia Eng.*, vol. 15, pp. 3341–3345, 2011. doi: 10.1016/j.proeng.2011.08.626.
[113] C. Jian, Y. Wang, M. Tao, and M. Zhang, "Time-constrained workflow scheduling in cloud environment using simulation annealing algorithm," *J. Eng. Sci. Technol. Rev.*, vol. 6, no. 5, pp. 33–37, 2013.
[114] S. S. Kim, J. H. Byeon, H. Liu, A. Abraham, and S. McLoone, "Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization. soft computing," *Soft Comput.*, vol. 17, pp. 867–882, 2013. doi: 10.1007/s00500-012-0957-7.
[115] Z. Fan, H. Shen, Y. Wu, and Y. Li, "Simulated-annealing load balancing for resource allocation in cloud environments," in *2013 Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, Taipei, Taiwan, IEEE, 2013. doi: 10.1109/PDCAT.2013.7.
[116] N. D. Lagaros, "An efficient dynamic load balancing algorithm," *Comput. Mech.*, vol. 53, no. 1, pp. 59–76, 2014. doi: 10.1007/s00466-013-0892-1.
[117] M. Rana, S. Bilgaiyan, and U. Kar, "A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms," in *2014 Int. Conf. Control, Instrum., Commun. Comput. Technol. (ICCICCT)*, Kanyakumari, India, IEEE, 2014. doi: 10.1109/ICCICCT.2014.6992964.
[118] F. Farahnakian *et al.*, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Trans. Serv. Comput.*, vol. 8, no. 2, pp. 187–198, 2014. doi: 10.1109/TSC.2014.2382555.

[119] R. Raju, J. Amudhavel, N. Kannan, and M. Monisha, "A bio inspired energy-aware multi objective chiropteran algorithm (EAMOCA) for hybrid cloud computing environment," in *2014 Int. Conf. Green Comput. Commun. Electr. Eng. (ICGCCEE)*, Coimbatore, India, IEEE, 2014. doi: 10.1109/ICGCCEE.2014.6922463.

[120] S. Xue, M. Li, X. Xu, J. Chen, and S. Xue, "An ACO-LB algorithm for task scheduling in the cloud environment," *J. Softw.*, vol. 9, no. 2, pp. 466–473, 2014. doi: 10.4304/jsw.9.2.265-273.

[121] S. Chitra, B. Madhusudhanan, G. R. Sakthidharan, and P. Saravanan, "Local minima jump PSO for workflow scheduling in cloud computing environments," in *Advances in Computer Science and Its Applications*. Berlin, Heidelberg: Springer, 2014, pp. 1225–1234. doi: 10.1007/978-3-642-41674-3_170.

[122] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egypt. Inform. J.*, vol. 16, no. 3, pp. 275–295, 2015. doi: 10.1016/j.eij.2015.07.001.

[123] W. T. Wen, C. D. Wang, D. S. Wu, and Y. Y. Xie, "An ACO-based scheduling strategy on load balancing in cloud computing environment," in *2015 Ninth Int. Conf. Front. Comput. Sci. Technol.*, Dalian, China, IEEE, 2015. doi: 10.1109/FCST.2015.41.

[124] K. M. Cho, P. W. Tsai, C. W. Tsai, and C. S. Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing," *Neural Comput. Appl.*, vol. 26, no. 6, pp. 1297–1309, 2015. doi: 10.1007/s00521-014-1804-9.

[125] M. Shojafar, S. Javanmardi, S. Abolfazli, and N. Cordeschi, "FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method," *Cluster Comput.*, vol. 18, no. 2, pp. 829–844, 2015. doi: 10.1007/s10586-014-0420-x.

[126] L. Sliwko and V. Getov, "A meta-heuristic load balancer for cloud computing systems," in *2015 IEEE 39th Annu. Comput. Softw. Appl. Conf.*, Taichung, Taiwan, IEEE, 2015. doi: 10.1109/COMPSAC.2015.223.

[127] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (GA) based load balancing strategy for cloud computing," *Procedia Technol.*, vol. 10, pp. 340–347, 2013. doi: 10.1016/j.protcy.2013.12.369.

[128] A. A. S. Farrag, S. A. Mahmoud, and M. El Sayed, "Intelligent cloud algorithms for load balancing problems: A survey," in *2015 IEEE Seventh Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Cairo, Egypt, IEEE, 2015. doi: 10.1109/IntelCIS.2015.7397223.

[129] I. Davydov and Y. Kochetov, "VNS-based heuristic with an exponential neighborhood for the server load balancing problem," *Electron. Notes in Discrete Math.*, vol. 47, pp. 53–60, 2015. doi: 10.1016/j.endm.2014.11.008.

[130] A. A. Beegom and M. Rajasree, "Genetic algorithm framework for bi-objective task scheduling in cloud computing systems," in *Int. Conf. Distrib. Comput. Internet Technol.*, Cham, Springer, 2015. doi: 10.1007/978-3-319-14977-6_38.

[131] K. Pan and J. Chen, "Load balancing in cloud computing environment based on an improved particle swarm optimization," in *2015 6th IEEE Int. Conf. Softw. Eng. Serv. Sci. (ICSESS)*, Beijing, China, IEEE, 2015. doi: 10.1109/ICSESS.2015.7339128.

[132] S. Aslanzadeh and Z. Chaczko, "Load balancing optimization in cloud computing: Applying Endocrine-particale swarm optimization," in *2015 IEEE Int. Conf. Electro/Inform. Technol. (Eit)*, Dekalb, IL, USA, IEEE, 2015. doi: 10.1109/EIT.2015.7293424.

[133] J. Gąsior and F. Seredyński, "Metaheuristic approaches to multiobjective job scheduling in cloud computing systems," in *2016 IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Luxembourg, IEEE, 2016. doi: 10.1109/CloudCom.2016.0046.

[134] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 66, pp. 64–82, 2016. doi: 10.1016/j.jnca.2016.01.018.

[135] N. R. Sabar, A. Song, and M. Zhang, "A variable local search based memetic algorithm for the load balancing problem in cloud computing," in *Eur. Conf. Appl. Evol. Comput.* Springer, 2016, vol. 66, pp. 64–82. doi: 10.1007/978-3-319-31204-0_18.

[136] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Glowworm swarm optimisation algorithm for virtual machine placement in cloud computing," in *Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud and Big Data Comput., Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, IEEE, 2016, pp. 808–814. doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0129.

[137] S. H. H. Madni, M. S. A. Latiff, and Y. Coulibaly, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 68, pp. 173–200, 2016. doi: 10.1016/j.jnca.2016.04.016.

[138] S. S. Rajput and V. S. Kushwah, "A genetic based improved load balanced min-min task scheduling algorithm for load balancing in cloud computing," in *2016 8th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Tehri, India, IEEE, 2016. doi: 10.1109/CICN.2016.139.

[139] J. Acharya, M. Mehta, and B. Sainim, "Particle swarm optimization based load balancing in cloud computing," in *2016 Int. Conf. on Commun. Electron. Syst. (ICCES)*, Coimbatore, India, IEEE, 2016. doi: 10.1109/CESYS.2016.7889943.

[140] M. Agnihotri and S. Sharma, "Execution analysis of load balancing particle swarm optimization algorithm in cloud data center," in *2016 Fourth Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Waknaghat, India, IEEE, 2016. doi: 10.1109/PDGC.2016.7913206.

[141] A. Dave, B. Patel, G. Bhatt, and Y. Vora, "Load balancing in cloud computing using particle swarm optimization on Xen Server," in *2017 Nirma Univ. Int. Conf. Eng. (NUiCONE)*, Ahmedabad, India, IEEE, 2017. doi: 10.1109/NUICONE.2017.8325618.

[142] M. Gamal, R. Rizk, H. Mahdi, and B. Elhady, "Bio-inspired load balancing algorithm in cloud computing," in *Int. Conf. Adv. Intell. Syst. Inf.*, Cham, Springer, 2017, vol. 639, pp. 579–589. doi: 10.1007/978-3-319-64861-3_54.

[143] S. Mousavi, A. Mosavi, and A. R. Varkonyi-Koczy, "A load balancing algorithm for resource allocation in cloud computing," in *Int. Conf. Glob. Res. Edu.*, Cham, Springer, 2017, vol. 660, pp. 289–296. doi: 10.1007/978-3-319-67459-9_36.

[144] G. Rjoub and J. Bentahar, "Cloud task scheduling based on swarm intelligence and machine learning," in *2017 IEEE 5th Int. Conf. Future Internet of Things and Cloud (FiCloud)*, Prague, Czech Republic, IEEE, 2017. doi: 10.1109/FiCloud.2017.52.

[145] A. Gupta and R. Garg, "Load balancing based task scheduling with ACO in cloud computing," in *2017 Int. Conf. Comput. Appl. (ICCA)*, Doha, Qatar, IEEE, 2017. doi: 10.1109/COMAPP.2017.8079781.

[146] A. Thakur and M. S. Goraya, "A taxonomic survey on load balancing in cloud," *J. Netw. Comput. Appl.*, vol. 98, pp. 43–57, 2017. doi: 10.1016/j.jnca.2017.08.020.

[147] A. M. Nilesh and C. A. Patel, "Load balancing in cloud computing using ant colony optimization," *Int. J. Comput. Eng. Technol.*, vol. 8, no. 6, pp. 54–59, 2017.

[148] M. Lagwal and N. Bhardwaj, "Load balancing in cloud computing using genetic algorithm," in *2017 Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, IEEE, 2017. doi: 10.1109/ICCONS.2017.8250524.

[149] A. Tripathi, S. Shukla, and D. Arora, "A hybrid optimization approach for load balancing in cloud computing," in *Advances in Computer and Computational Sciences*. Singapore, Springer, 2018, vol. 554, pp. 197–206, doi: 10.1007/978-981-10-3773-3_19.

[150] M. Hanine, "QoS in the cloud computing: A load balancing approach using simulated annealing algorithm," in *Int. Conf. Big Data, Cloud and Appl.*, Cham, Springer, 2018, vol. 872, pp. 43–54. doi: 10.1007/978-3-319-96292-4_4.

[151] A. Belgacem, K. Beghdad-Bey, and H. Nacer, "Task scheduling in cloud computing environment: A comprehensive analysis," in *Int. Conf. Comput. Sci. Appl.*, Cham, Springer, 2018, vol. 50, pp. 14–26. doi: 10.1007/978-3-319-98352-3_3.

[152] B. N. Gohil and D. R. Patel, "An improved grey wolf optimizer (iGWO) for load balancing in cloud computing environment," in *Int. Conf. Algorithms and Archit. Parallel Process.*, Cham, Springer, 2018, vol. 11338, pp. 3–9. doi: 10.1007/978-3-030-05234-8_1.

[153] S. Dam, G. Mandal, K. Dasgupta, and P. Dutta, "An ant-colony-based meta-heuristic approach for load balancing in cloud computing," in *Applied Computational Intelligence and Soft Computing in Engineering, IGI Global*, IGI Global, 2018, pp. 204–232, doi: 10.4018/978-1-5225-3129-6.ch009.

[154] D. Garg and P. Kumar, "A Survey on Metaheuristic approaches and its evaluation for load balancing in cloud computing," in *Int. Conf. Adv. Inf. Comput. Res.*, Singapore, Springer, 2018, vol. 955, pp. 585–599. doi: 10.1007/978-981-13-3140-4_53.

[155] J. Zhou and S. Dong, "Hybrid glowworm swarm optimization for task scheduling in the cloud environment," *Eng. Optim., Taylor & Francis*, vol. 50, no. 6, pp. 949–964, 2018. doi: 10.1080/0305215X.2017.1361418.

[156] F. Luo, Y. Yuan, W. Ding, and H. Lu, "An improved particle swarm optimization algorithm based on adaptive weight for task scheduling in cloud computing," in *Proc. 2nd Int. Conf. Comput. Sci. Appl. Eng.*, Hohhot, China, 2018, pp. 1–5. doi: 10.1145/3207677.3278089.

[157] H. Alazzam, A. Alsmady, W. Mardini, and A. Enizat, "Load balancing in cloud computing using water flow-like algorithm," in *Proc. Second Int. Conf. Data Sci., E-Learn. Inf. Syst.*, Dubai, United Arab Emirates, 2019, pp. 1–6. doi: 10.1145/3368691.3368720.

[158] M. M. Golchi, S. Saraeian, and M. Heydari, "A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation," *Comput. Netw.*, vol. 162, pp. 106860, 2019. doi: 10.1016/j.comnet.2019.106860.

[159] B. Jana, M. Chakraborty, and T. Mandal, "A task scheduling technique based on particle swarm optimization algorithm in cloud environment," in *Soft Computing: Theories and Applications*. Singapore: Springer, pp. 525–536, 2019.

[160] A. A. S. Farrag, S. A. Mohamad, and M. El Sayed, "Swarm optimization for solving load balancing in cloud computing," in *Int. Conf. Adv. Mach. Learn. Technol. Appl.,*, Cham, Springer, 2019, pp. 102–113. doi: 10.1007/978-3-030-14118-9_11.

[161] N. Mansouri, B. M. H. Zade, and M. M. Javidi, "Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory," *Comput. Ind. Eng.*, vol. 130, pp. 597–633, 2019. doi: 10.1016/j.cie.2019.03.006.

[162] W. Li, T. Jian, Y. Wang, and X. Ma, "Research on virtual machine load balancing based on improved particle swarm optimization," in *2019 IEEE Symp. Series Comput. Intell. (SSCI)*, Xiamen, China, IEEE, 2019, pp. 2846–2852. doi: 10.1109/SSCI44817.2019.9002730.

[163] S. Afzal and G. Kavitha, "Load balancing in cloud computing-A hierarchical taxonomical classification," *J. Cloud Comput.*, vol. 8, no. 1, pp. 1–24, 2019. doi: 10.1186/s13677-019-0146-7.

[164] A. Kaur and B. Kaur, "Load balancing optimization based on hybrid Heuristic-Metaheuristic techniques in cloud environment," *J. King Saud Univ.–Comput. Inf. Sci.*, vol. 34, no. 3, pp. 813–824, 2019. doi: 10.1016/j.jksuci.2019.02.010.

[165] J. P. B. Mapetu, Z. Chen, and L. Kong, "Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing," *Appl. Intell.*, vol. 49, no. 9, pp. 3308–3330, 2019. doi: 10.1007/s10489-019-01448-x.

[166] R. Valarmathi and T. Sheela, "Ranging and tuning based particle swarm optimization with bat algorithm for task scheduling in cloud computing," *Cluster Comput.*, vol. 22, no. 5, pp. 11975–11988, 2019. doi: 10.1007/s10586-017-1534-8.

[167] A. F. S. Devaraj, S. D. Mohamed Elhoseny, E. L. Lydia, and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *J. Parallel Distr. Comput.*, vol. 142, pp. 36–45, 2020. doi: 10.1016/j.jpdc.2020.03.022.

[168] R. Agarwal, N. Baghel, and M. A. Khan, "Load balancing in cloud computing using mutation based particle swarm optimization," in *2020 Int. Conf. Contemp. Comput. Appl. (IC3A)*, Lucknow, India, IEEE, 2020. doi: 10.1109/IC3A48958.2020.233295.

[169] M. Junaid *et al.*, "Modeling an optimized approach for load balancing in cloud," *IEEE Access*, vol. 8, pp. 173208–173226, 2020. doi: 10.1109/ACCESS.2020.3024113.

[170] K. Bhushan, "Load balancing in cloud through task scheduling," in *Recent Trends in Communication and Intelligent Systems*. Singapore: Springer, 2020, pp. 195–204. doi: 10.1007/978-981-15-0426-6_21.
[171] M. Junaid, A. Sohail, A. Ahmed, A. Baz, I. A. Khan and H. Alhakami, "A hybrid model for load balancing in cloud using file type formatting," *IEEE Access*, vol. 8, pp. 118135–118155, 2020. doi: 10.1109/ACCESS.2020.3003825.
[172] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2477–2489, 2021. doi: 10.11591/ijece.v11i3.pp2477-2489.
[173] K. Balaji, P. S. Kiran, and M. S. Kumar, "An energy efficient load balancing on cloud computing using adaptive cat swarm optimization," *Mater. Today: Proc.*, 2021. doi: 10.1016/j.matpr.2020.11.106.
[174] A. Muteeh, M. Sardaraz, and M. Tahir, "MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization," *Cluster Comput.*, vol. 24, pp. 3135–3145, 2021. doi: 10.1007/s10586-021-03322-3.
[175] H. Singh, S. Tyagi, and P. Kumar, "Cloud resource mapping through crow search inspired metaheuristic load balancing technique," *Comput. Electr. Eng.*, vol. 93, pp. 107221, 2021. doi: 10.1016/j.compeleceng.2021.107221.
[176] R. Kaviarasan, P. Harikrishna, and A. Arulmurugan, "Load balancing in cloud environment using enhanced migration and adjustment operator based monarch butterfly optimization," *Adv. Eng. Softw.*, vol. 169, pp. 103128, 2022. doi: 10.1016/j.advengsoft.2022.103128.
[177] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends," *Swarm Evol. Comput.*, vol. 62, pp. 100841, 2021. doi: 10.1016/j.swevo.2021.100841.
[178] A. Thakur and M. S. Goraya, "RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment," *Simul. Model. Pract. Theory*, vol. 116, pp. 102485, 2022. doi: 10.1016/j.simpat.2021.102485.
[179] S. Sefati, M. Mousavinasab, and R. Z. Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: Performance evaluation," *J. Supercomput.*, vol. 78, no. 1, pp. 18–42, 2022. doi: 10.1007/s11227-021-03810-8.
[180] Y. Xu and K. Abnoosian, "A new metaheuristic-based method for solving the virtual machines migration problem in the green cloud computing," *Concurr. Comput.*, vol. 34, no. 3, pp. e6579, 2022. doi: 10.1002/cpe.6579.
[181] M. O. Al-Wesabi, M. Obayya, M. A. Hamza c, J. S. Alzahrani, D. Gupta, and S. Kumar, "Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment," *Sustain. Comput.: Inf. Syst.*, vol. 35, pp. 100686, 2022. doi: 10.1016/j.suscom.2022.100686.
[182] S. M. Mirmohseni, C. Tang, and A. Javadpour, "FPSO-GA: A fuzzy metaheuristic load balancing algorithm to reduce energy consumption in cloud networks," *Wirel. Pers. Commun.*, vol. 127, pp. 2799–2821, 2022. doi: 10.1007/s11277-022-09897-3.
[183] K. Suresh *et al.*, "Enhanced metaheuristic algorithm-based load balancing in a 5G cloud radio access network," *Electronics*, vol. 11, no. 21, pp. 3611, 2022. doi: 10.3390/electronics11213611.
[184] B. Prabhakara, C. Naikodi, and L. Suresh, "Hybrid meta-heuristic technique load balancing for cloud-based virtual machines," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1, pp. 132–139, 2023.
[185] J. P. Gabhane, S. Pathak, and N. M. Thakare, "A novel hybrid multi-resource load balancing approach using ant colony optimization with Tabu search for cloud computing," *Innov. Syst. Softw. Eng.*, vol. 19, no. 1, pp. 81–90, 2023. doi: 10.1007/s11334-022-00508-9.
[186] N. Marathe, A. Gandhi, and J. M. Shah, "Docker swarm and kubernetes in cloud computing environment," in *3rd Int. Conf. Trends in Electron. Inf. (ICOEI)*, *IEEE*, Tirunelveli, India, 2019, pp. 179–184. doi: 10.1109/ICOEI.2019.8862654.
[187] A. Modak, S. D. Chaudhary, P. S. Paygude, and S. R. Ldate, "Techniques to secure data on cloud: Docker swarm or kubernetes?" in *2018 Second Int. Conf. Inventive Commun. Comput. Technol.(ICICCT)*, *IEEE*, Coimbatore, India, 2018, pp. 7–12. doi: 10.1109/ICICCT.2018.8473104.

[188] A. Saadat and E. Masehian, "Load balancing in cloud computing using genetic algorithm and fuzzy logic," in *2019 Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, NV, USA, IEEE, 2019, pp. 1435–1440. doi: 10.1109/CSCI49370.2019.00268.

[189] R. M. Alguliyev, Y. N. Imamverdiyev, and F. J. Abdullayeva, "PSO-based load balancing method in cloud computing," *Autom. Control Comput. Sci.*, vol. 53, no. 1, pp. 45–55, 2019. doi: 10.3103/S0146411619010024.

[190] K. V. Kumar and A. Rajesh, "Multi-objective load balancing in cloud computing: A meta-heuristic approach," *Cybern. Syst.*, vol. 54, no. 8, pp. 1466–1493, 2023. doi: 10.1080/01969722.2022.2145656.

[191] P. Geetha, S. J. Vivekanandan, R. Yogitha, and M. S. Jeyalakshmim, "Optimal load balancing in cloud: Introduction to hybrid optimization algorithm," *Expert. Syst. Appl.*, vol. 237, pp. 121450, 2024. doi: 10.1016/j.eswa.2023.121450.

[192] S. Ghafir, M. A. Alam, F. Siddiqui, and S. Naaz, "Load balancing in cloud computing via intelligent PSO-based feedback controller," *Sustain. Comput. Inf. Syst.*, vol. 41, pp. 100948, 2024. doi: 10.1016/j.suscom.2023.100948.

[193] S. Simaiya *et al.*, "A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques," *Sci. Rep.*, vol. 14, no. 1, pp. 1337, 2024. doi: 10.1038/s41598-024-51466-0.

[194] M. H. Nebagiri and L. P. Hnumanthappa, "Multi-Objective of load balancing in cloud computing using cuckoo search optimization based simulation annealing," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 9s, pp. 466–474, 2024.

[195] D. C. Devi and V. R. Uthariaraj, "Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks," *Sci. World J.*, vol. 2016, 2016. doi: 10.1155/2016/3896065.

[196] S. A. Ali and M. Alam, "Resource-aware min-min (RAMM) algorithm for resource allocation in cloud computing environment," 2018. doi: 10.48550/arXiv.1803.00045.

[197] A. K. Moses, A. J. Bamidele, O. R. Oluwaseun, S. Misra, and A. A. Emmanuela, "Applicability of MMRR load balancing algorithm in cloud computing," *Int. J. Comput. Math. Comput. Syst. Theory*, vol. 6, no. 1, pp. 7–20, 2021. doi: 10.1080/23799927.2020.1854864.

[198] M. Ibrahim *et al.*, "A comparative analysis of task scheduling approaches in cloud computing," in *2020 20th IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput. (CCGRID)*, Melbourne, VIC, Australia, IEEE, 2020. doi: 10.1109/CCGrid49817.2020.00-23.

[199] M. L. Chiang, H. C. Hsieh, Y. H. Cheng, W. L. Lin, and B. H. Zeng, "Improvement of tasks scheduling algorithm based on load balancing candidate method under cloud computing environment," *Expert. Syst. Appl.*, vol. 212, pp. 118714, 2023. doi: 10.1016/j.eswa.2022.118714.

[200] D. B. LD and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, 2013. doi: 10.1016/j.asoc.2013.01.025.

[201] A. Ragmani, A. E. Omri, N. Abghour, K. Moussaid, and M. Rida, "A performed load balancing algorithm for public cloud computing using ant colony optimization," *Recent Pat. Comput. Sci.*, vol. 11, no. 3, pp. 179–195, 2018. doi: 10.2174/2213275911666180903124609.

[202] A. K. Duggal and M. Dave, "A comparative study of load balancing algorithms in a cloud environment," in *Adv. Comput. Intell. Syst.*, Singapore, Springer, 2020, pp. 115–126. doi: 10.1007/978-981-15-0222-4_10.

[203] V. Mohammadian, N. J. Navimipour, M. Hosseinzadeh, and A. Darwesh, "LBAA: A novel load balancing mechanism in cloud environments using ant colony optimization and artificial bee colony algorithms," *Int. J. Commun. Syst.*, vol. 36, no. 9, pp. e5481, 2023. doi: 10.1002/dac.5481.

[204] G. Muthsamy and S. R. Chandran, "Task scheduling using artificial bee foraging optimization for load balancing in cloud data centers," *Comput. Appl. Eng. Educ.*, vol. 28, no. 4, pp. 769–778, 2020. doi: 10.1002/cae.22236.

[205] D. Patel, M. K. Patra, and B. Sahoo, "GWO Based task allocation for load balancing in containerized cloud," in *Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, IEEE, 2020, pp. 655–659. doi: 10.1109/ICICT48043.2020.9112525.

[206] A. Thakur and M. S. Goraya, "RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment," *Simul. Model. Pract. Theory*, vol. 116, pp. 102485, 2022. doi: 10.1016/j.simpat.2021.102485.
[207] X. Chen *et al.*, "A WOA-based optimization approach for task scheduling in cloud computing systems," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3117–3128, 2020. doi: 10.1109/JSYST.2019.2960088.
[208] R. Gulbaz, A. B. Siddiqui, N. Anjum, A. A. Alotaibi, T. Althobaiti and N. Ramzan, "Balancer genetic algorithm—A novel task scheduling optimization approach in cloud computing," *Appl. Sci.*, vol. 11, no. 14, pp. 6244, 2021. doi: 10.3390/app11146244.
[209] C. Li, J. Tang, and Y. Luo, "Service cost-based resource optimization and load balancing for edge and cloud environment," *Knowl. Inf. Syst.*, vol. 62, pp. 4255–4275, 2020. doi: 10.1007/s10115-020-01489-6.
[210] E. Ilavarasan and P. Thambidurai, "Low complexity performance effective task scheduling algorithm for heterogeneous computing environments," *J. Comput. Sci.*, vol. 3, no. 2, pp. 94–103, 2007.
[211] U. Jena, P. Das, and M. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment," *J. King Saud Univ.–Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2332–2342, 2022. doi: 10.1016/j.jksuci.2020.01.012.
[212] T. C. Hung and N. X. Phi, "Study the effect of parameters to load balancing in cloud computing," *Ratio*, vol. 45, pp. 82–29, 2016. doi: 10.5121/ijcnc.2016.8303.
[213] S. Afzal and G. Kavitha, "A taxonomic classification of load balancing metrics: A systematic review," in *Proc. 33rd Indian Eng. Congr.*, Udaipur, India, 2019, pp. 85–90.
[214] B. N. Gohil and D. R. Patel, "A hybrid GWO-PSO algorithm for load balancing in cloud computing environment," in *2018 Second Int. Conf. Green Comput. Internet of Things (ICGCIoT)*, Bangalore, India, IEEE, 2018. doi: 10.1109/ICGCIoT.2018.8753111.
[215] G. Natesan and A. Chokkalingam, "An improved grey wolf optimization algorithm based task scheduling in cloud computing environment," *Int. Arab J. Inf. Technol.*, vol. 17, no. 1, pp. 73–81, 2020. doi: 10.34028/iajit/17/1/9.
[216] A. Ullah, N. M. Nawi, and M. H. Khan, "BAT algorithm used for load balancing purpose in cloud computing: An overview," *Int. J. High Perform. Comput. Netw.*, vol. 16, no. 1, pp. 43–54, 2020. doi: 10.1504/IJHPCN.2020.110258.
[217] S. Tiwari and C. Bhatt, "A Comprehensive study on cloud computing: Architecture, load balancing, task scheduling and meta-heuristic optimization," in *Intelligent Cyber Physical Systems and Internet of Things*. Cham: Springer, 2023, pp. 137–162, doi: 10.1007/978-3-031-18497-0_11.
[218] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," *ACM Comput. Surv. (CSUR)*, vol. 51, no. 6, pp. 1–35, 2019. doi: 10.1145/3281010.