frontiers | Frontiers in Computer Science

TYPE Systematic Review
PUBLISHED 08 July 2024
DOI 10.3389/fcomp.2024.1288552

OPEN ACCESS

EDITED BY
Saif Ul Islam,
Institute of Space Technology, Pakistan

REVIEWED BY
Kashav Ajmera,
Jaypee Institute of Information Technology, India
Muhammad Anwar,
University of Education Lahore, Pakistan

*CORRESPONDENCE
Wagdi Alrawagfeh
✉ wagdi.alrawagfeh@udst.edu.qa

RECEIVED 04 September 2023
ACCEPTED 27 May 2024
PUBLISHED 08 July 2024

CITATION
Rana N, Jeribi F, Khan Z, Alrawagfeh W, Ben Dhaou I, Haseebuddin M and Uddin M (2024) A systematic literature review on contemporary and future trends in virtual machine scheduling techniques in cloud and multi-access computing.
Front. Comput. Sci. 6:1288552.
doi: 10.3389/fcomp.2024.1288552

A systematic literature review on contemporary and future trends in virtual machine scheduling techniques in cloud and multi-access computing

Nadim Rana[1], Fathe Jeribi[1], Zeba Khan[2], Wagdi Alrawagfeh[3]*, Imed Ben Dhaou[4,5], Mohammad Haseebuddin[1] and Mueen Uddin[3]

[1]College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia, [2]Department of Computer and Information, Applied College, Jazan University, Jazan, Saudi Arabia, [3]College of Computing and Information Technology, University of Doha for Science and Technology, Doha, Qatar, [4]Department of Computer Science, Hekma School of Engineering, Computing and Informatics, Dar Al-Hekma University, Jeddah, Saudi Arabia, [5]Department of Computing, University of Turku, Turku, Finland

Introduction: The migration of business and scientific operations to the cloud and the surge in data from IoT devices have intensified the complexity of cloud resource scheduling. Ensuring efficient resource distribution in line with user-specified SLA and QoS demands novel scheduling solutions. This study scrutinizes contemporary Virtual Machine (VM) scheduling strategies, shedding light on the complexities and future prospects of VM design and aims to propel further research by highlighting existing obstacles and untapped potential in the ever-evolving realm of cloud and multi-access edge computing (MEC).

Method: Implementing a Systematic Literature Review (SLR), this research dissects VM scheduling techniques. A meticulous selection process distilled 67 seminal studies from an initial corpus of 722, spanning from 2008 to 2022. This critical filtration has been pivotal for grasping the developmental trajectory and current tendencies in VM scheduling practices.

Result: The in-depth examination of 67 studies on VM scheduling has produced a taxonomic breakdown into three principal methodologies: traditional, heuristic, and meta-heuristic. The review underscores a marked shift toward heuristic and meta-heuristic methods, reflecting their growing significance in the advancement of VM scheduling.

Conclusion: Although VM scheduling has progressed markedly, the focus remains predominantly on metaheuristic and heuristic approaches. The analysis enlightens ongoing challenges and the direction of future developments, highlighting the necessity for persistent research and innovation in this sector.

KEYWORDS

cloud computing, virtualization, SLA, virtual machine scheduling, QoS, internet of things, multi-access computing

1 Introduction

Virtual machine (VM) scheduling in cloud computing refers to the process of assigning virtual machines to physical servers in a way that optimizes the utilization of resources, complies with service level agreements (SLA), and ensures the quality of service (QoS). The goal is to manage the computing resources efficiently, handling tasks such as load balancing, reducing energy consumption, and minimizing response times (Beloglazov and Buyya, 2012; Xu et al., 2017; Sayadnavard et al., 2022). VM scheduling can be categorized into two: Static and Dynamic. Static VM scheduling typically involves pre-determined placement of virtual machines on physical hosts, often based on initial load estimates and without consideration for changing workloads. Dynamic VM scheduling, conversely, adapts to real-time conditions, continuously optimizing resource allocation as demands fluctuate. Dynamic VM scheduling emphasizes real-time optimization techniques that are characteristic of dynamic approaches, which adapt to ongoing changes in the cloud environment to improve efficiency and performance. This Systematic Literature Review (SLR) primarily focuses on dynamic VM scheduling (Ajmera and Tewari, 2023; Ajmera and Kumar Tewari, 2024). As a consequence of the advancement of cloud computing, many computing resources are provisioned as utilities on a metered basis to the client over the Internet (Buyya and Ranjan, 2010; Manvi and Shyam, 2014). Based on user demand, the cloud provider may easily and dynamically allocate and release these resources (Li W. et al., 2017). Virtual Machines (VMs) play the most critical role in the virtual cloud environment as a resource container with business services encapsulated. As a matter of fact, due to ever-changing conditions, VM scheduling and optimization in a heterogeneous environment remains a challenging issue for cloud resource providers (Khosravi et al., 2017). From the perspective of cloud providers, a massive number of resources are provisioned on VMs. In the cloud, thousands of users share the same amount of available resources fairly and dynamically. VM scheduling, at the same time, aims at ensuring the Quality of Service (QoS) along with cost-effectiveness (Qi et al., 2020). Some major issues that supposedly interconnected with Infrastructure-as-a-Service (IaaS) in cloud computing are the resource organization (Mustafa et al., 2015), data management (Wang et al., 2018), network infrastructure management (Ahmad et al., 2017), virtualization and multi-tenancy (Duan and Yang, 2017), application-programming-interfaces (APIs), interoperability (Challita et al., 2017), VM security (Uddin et al., 2015; Aikat et al., 2017) and the load-balancing (Mousavi et al., 2018).

VM scheduling ensures a balancing scenario in which VMs are allocated to the available Physical Machines (PMs) as per resource requirements (Shaw and Singh, 2014). Moreover, VM scheduling techniques are utilized to schedule VM requests of particular data center (DC), according to the required computing resources. In essence, the optimization of VM scheduling techniques to achieve efficient and effective resource scheduling gained larger attention of researchers in cloud computing (Rodriguez and Buyya, 2017).

The present literature in cloud computing scheduling can be categorized using performance matrix and scheduling methods. The surveys that are based on performance focus on specific issues such as (i) energy-aware scheduling, (ii) cost-aware scheduling, (iii) load balancing-aware scheduling, and (iv) utilization-aware scheduling. The method-based survey categorizes (i) VM allocation, (ii) VM consolidation or placement, (iii) VM migration, (iv) VM provisioning, and (v) VM scheduling. The classification, as mentioned above, is discussed in Section 4 of this study. According to the author's best knowledge, several polls and studies have been conducted on the themes that were discussed earlier. However, an extensive study on VM scheduling has been found missing in the available cloud computing literature. Hence, this study tries to do an extensive systematic survey on VM scheduling and presents the following contributions:

- To provide the outline of the techniques in VM scheduling in the same manner as these techniques have been applied in cloud computing.
- To present syntheses of contemporary issues and challenges and mention the problems related to VM scheduling.
- To present a comparative analysis of VM scheduling methods and parameters in cloud and multi-access computing (MAC).
- To evaluate various VM scheduling approaches critically while highlighting their drawbacks and advantages.
- To emphasize the importance of VM scheduling as a baseline for researchers to solve issues in near future.

Extensive examination and analysis of existing literature on contemporary issues and research gaps are crucial for generating ideas. This study tries to disseminate the most relevant VM scheduling techniques and approaches available in the literature and anticipates that they can effectively improve modern VM scheduling methods. This study attempts to present recent trends, requirements, and future scopes in the development of VM scheduling techniques in cloud computing.

The structure of the study is organized as follows: Section 2 discusses literature reviews in cloud scheduling. Section 3 presents the research methodology. Section 4 illustrates VM management methods and systems models. Section 5 presents the analysis of VM scheduling approaches and their parameters. Section 6 discusses scheduling in mobile edge computing and the validity of the research. Finally, Section 7 illustrates research issues and opportunities. Section 8 concludes with the findings of the literature review.

2 Literature review

Numerous studies are presented in the area of cloud scheduling, and some generic challenges are discussed, such as resource scheduling, resource provisioning, and load balancing. However, current studies have no extensive systematic survey on VM scheduling. This section refers to some studies in the area of cloud scheduling. When allocating dynamic, heterogeneous, and shared resources, resource scheduling in cloud environments is considered one of the most crucial challenges. To provide reliable and cost-effective access, overloading of those resources must be prevented by proper load balancing and effective scheduling techniques.

TABLE 1 Summary of previous literature in VM scheduling.

Previous reviews	VM scheduling	Problem formulation	Classification of VM scheduling	Parametric analysis	Simulation tool and environment	Dataset available	Architecture	Period covered
Li et al. (2013)							√	2002–2009
Beloglazov et al. (2012)	√	√		√		√	√	1991–2012
Rathore and Chana (2014)				√				1999–2014
Xu X. et al. (2018)		√	√	√			√	2003–2013
Kumar et al. (2019)				√	√		√	2009–2014
Kalra and Singh (2015)			√	√				2001–2005
Zhan et al. (2015)	√		√		√		√	2003–2014
Ahmad et al. (2015a)	√		√	√	√		√	1993–2014
Ahmad et al. (2015b)	√		√	√			√	1997–2015
Madni et al. (2016)			√	√				1954–2016
Madni et al. (2017)		√	√	√			√	2008–2016
Xu et al. (2017)			√	√			√	2008–2016
Our review	√	√	√	√	√	√	√	2008–2022

In the study mentioned in the reference (Kumar et al., 2019), a comprehensive survey of resource scheduling algorithms offer an analysis based on categorizing some parameters that include load balancing, energy management, and makespan. The study observed that there is no scheduling algorithm that has the potential to effectively address all parameters of VM scheduling. Furthermore, the study discussed some task scheduling algorithms, limitations, and future problems. However, the scope of the study is restricted to only grid computing. Similar study in references (Beloglazov et al., 2012; Rathore and Chana, 2014; Li H. et al., 2017; Uddin et al., 2021) presented a study of energy-aware resource allocation methods focusing on the QoS. They mentioned some critical and open challenges in resource scheduling, particularly energy management in a cloud data center. According to their analysis of previous studies, the challenges are enumerated as follows: (1) processes that are quick and energy-efficient for placing VMs and can anticipate workload peaks to prevent performance deprivation in a heterogeneous environment, (2) energy-based virtual network topology optimization technique among VMs for the best location to lessen network traffic congestion, (3) to properly regulate temperature and energy use and new heat management algorithms, (4) even workloads and workload-aware resource allocation processes, and (5) scalability and fault-tolerance techniques for VM placement (VMP) challenges that are decentralized and distributed.

In a similar type of study, Li et al. (2010) delved into VM scheduling issues within cloud data centers. They also offered an overview of contemporary technologies in the realm of cloud computing, encompassing aspects such as virtualization, resource allocation, VM migration, security measures, and performance evaluation. Parallel to this, they highlighted emerging challenges and complexities, including CPU design, resource governance, security maintenance strategies, and evaluation methods in multi-VM environments. Even with their research, their research fell short in areas such as structured categorization, problem definition, parametric study, and a comprehensive exploration of the techniques, as pointed out in previous studies.

Analyzing the cloud computing architecture, Zhan et al. (2015) systematically presented two-level taxonomy of cloud resources. Researchers have critically examined the issue and remedy of cloud scheduling in their review. Additionally, they investigated EC methodologies and talked about several cutting-edge evolutionary algorithms and their potential to solve the cloud scheduling issue. Based on their categorization, they have also identified some problems and research fields, such as distributed parallel scheduling, adaptive dynamic scheduling, large-scale scheduling, and multi-objective scheduling. They have also highlighted some of the most cutting-edge future themes, including the Internet of Things and the convergence of cyber and physical systems with big data. However, they should have described the problem's mathematical modeling or included any parametric analysis in the study.

In another investigation, Xu et al. (2013) described the causes of the performance overhead problem while scheduling virtual resources under several scenarios, i.e., from single server

virtualization to multiple server virtualization in distributed data centers. The review presents a detailed comparison of contemporary migration techniques and modeling approaches to manage performance overhead problems. However, the authors suggest that a lot remains to be resolved to ensure the predictable performance of VM with guaranteed SLA. Similarly, Madni et al. (2016) examine the difficulties and possibilities in resource scheduling for cloud infrastructure as a service (IaaS). They categorize the previous scheduling schemes according to the issues addressed and performance metrics and present a classification scheme. Furthermore, some essential parameters are evaluated and their strengths and weaknesses are highlighted. Finally, they suggest some innovative ideas for future enhancements in resource scheduling techniques.

Meta-heuristic strategies have set a standard in VM scheduling because they produce efficient and nearly optimal outcomes in a feasible time frame. Numerous studies have been conducted to evaluate the performance of these advanced meta-heuristic algorithms. In a parallel vein, Kalra and Singh (2015) examined six prominent meta-heuristic optimization methods, including Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), League Championship Algorithm (LCA), and the Bat algorithm. These techniques are presented within a taxonomic structure, with comparisons made based on several scheduling metrics such as task recognition, SLA observance, and energy consciousness. They further delve into the practical applications of these meta-heuristic methods and the existing challenges in grid or cloud scheduling. Nevertheless, their review is confined to specific meta-heuristic methods and optimization metrics.

In another development, Madni et al. (2017) investigated the potential of existing state-of-the-art Meta-heuristic techniques for resource scheduling in a cloud computing environment to maximize the cloud provider's financial benefit and minimize cost for cloud users. In their research, they selected 23 meta-heuristic technique studies between 1954 and 2015. They compared meta-heuristic techniques with traditional techniques to evaluate the performance criteria of the algorithms. They claimed that there can be several ways to enhance the performance of these algorithms which can further solve the resource scheduling problem. However, the focus of the study is only on meta-heuristic methods.

Unlike previous studies shown in Table 1, our research presents an extensive (not exhaustive) review of VM scheduling techniques and presents the most appropriate categorization, problem formulation, architecture, and future challenges. Then, based on our research, we formalize three questions and choose the most important study from the most trustworthy research database to address them. Furthermore, we delineate the importance of VM scheduling techniques, current issues and challenges, and future direction to support future research.

3 Research methodology

According to the guidelines mentioned in Kitchenham et al. (2007) and Kitchenham (2004), the presented SLR employs a tried-and-true procedure to examine earlier research by other researchers, which should provide sufficient details for other researchers to reproduce in the future (Charband and Navimipour, 2016; Navimipour and Charband, 2016). Following the best practice and guidelines, this study developed a protocol to accumulate the necessary details for VM scheduling techniques, approaches, and their parameters. Three research questions are established based on the analysis of collected literature on the main concerns with VM scheduling in cloud computing. The research questions are presented in the section below.

3.1 Research questions

In this section, the most important problems and challenges related to cloud-based scheduling were discussed, including resource provisioning, resource scheduling, task scheduling, VM scheduling, resource utilization, load balancing, and prospective balancing solutions. Therefore, the effort of this research is to address the following important research questions:

Research Question (RQ1): What is the significance of VM scheduling in light of the increase in cloud usage? RQ1 will try to survey several VM scheduling studies published over the period under study, to underline the importance of VM scheduling along with increasing cloud usage.

RQ2: How many of the current scheduling strategies achieve the primary VM scheduling goals concerning the particular parameters? RQ2's objective is to assess current VM scheduling strategies in a cloud computing system based on the key VM scheduling parameters.

RQ3: What problems and potential solutions were found concerning VM scheduling for upcoming research trends? RQ3's goal is to classify the difficulties in VM scheduling in cloud computing and the methods utilized to ensure QoS in the system.

The specific responses to the questions posed within the scope of this study are obtained through a multi-stage approach. Once the necessity for the research has been established, a standardized process has been used to frame the research topic. The research must go through several processes to adhere to the protocol, including the search request, source selection, quality assessment criteria, extraction, and information analysis approach.

For respected online academic libraries and databases, search strings or keywords were created by defining keywords, which are based on inclusion and exclusion criteria. The Boolean "OR" and Boolean "AND" operators are used to connect similar and alternative spellings for each of the question elements to define keywords (Milani and Navimipour, 2016). The search string is created using a combination of synonyms and alternate spellings for each element of the inquiry to find the pertinent topic. The best keywords from our subject have been chosen based on the established search string to obtain the desired outcome from databases. Thus, the terms "Virtual Machine," "VM," "Cloud," "Scheduling," and "Scheduler" have been chosen as the five keywords. The query was defined after going through many processes and assessing the findings of our preliminary study as a pilot to look at the result's coverage. Supposedly, if we used the pilot search from our studies and the original query did not yield

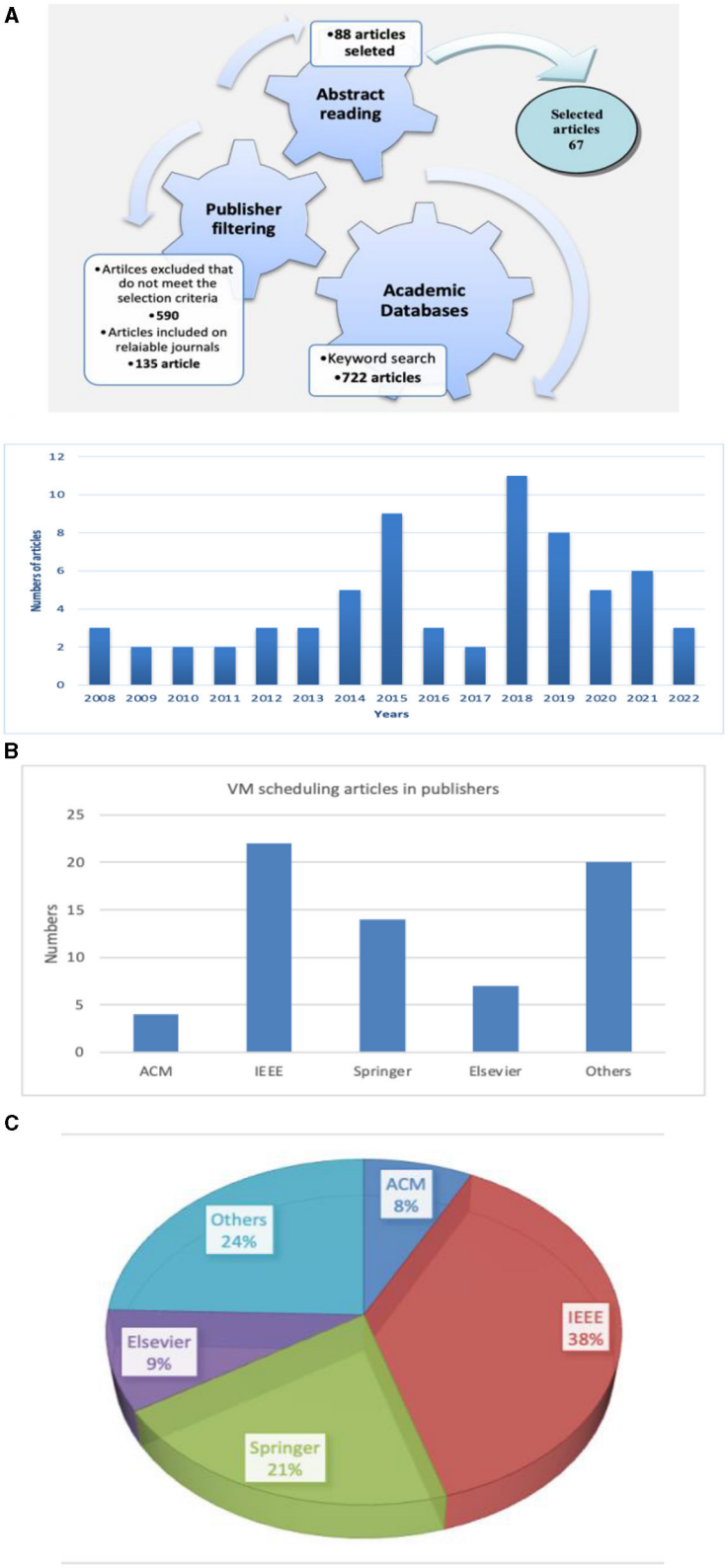

FIGURE 1
Virtual machine scheduling article identification and selection process. **(A)** Number of studies selected in each phase. **(B)** Number of articles present in publishers. **(C)** Percentage of articles in publisher out of 67 selected studies.

the required results, we then modified our search using terms such as "Virtual machine" OR "VM" AND "scheduling," OR "scheduler." The search was carried out in August 2018 and covered the years 2008 to 2022.

3.2 Selection of sources

In the process of article selection, we have chosen some of the most relevant journal articles and conference papers from the most relevant academic databases for our search query. Subsequently, the selected results have been classified based on the publishers. We have searched through Web of Science, Scopus, and Google Scholar as our primary data source search engines. As a result, practically, all of the articles published in the most reputable online journals and conferences that have undergone technical and scientific peer review were covered by the search process: Springer Link, ScienceDirect, IEEE Infocom, IEEE -Xplore, ACM-Digital Library, and ICDCS.

3.3 Selection criteria

An assessment method has been followed for the inclusion of the studies based on the prepared quality assessment checklist (QAC) in the study mentioned in the reference (Kitchenham et al., 2009), to assess only specific articles from the peer-reviewed journal published between 2008 and 2022 as mentioned in Figure 1. Based on the above filtering and analysis of the articles based on the checklist, a list of questions is prepared: (a) Does the research approach depend on the research article? (b) Is the research approach appropriate for the issue covered in the article? (c) Is the analysis of the study adequately done? (d) Does the survey meet the requirements for evaluation?

3.4 Extraction of data and quality evaluation process

We compile the data from the chosen research during the data extraction process for additional analysis. Primarily, we selected a sum of 722 articles from all relevant databases. Then, we read keywords, abstracts, and concepts that match our topic of study. Consequently, 88 articles were selected based on abstract, and the rest of the studies were discarded. Then, the full body of each article was studied; those studies were not found suitable as the details mentioned inside the text were also removed. After summarizing the studies based on inclusion and exclusion criteria and QAC, 67 articles were selected for our review. Figure 1 demonstrates the overall inclusion and exclusion process followed in this study to identify the most suitable articles. As per the analysis of the retrieved data from relevant sources, a significant amount of growth can be observed in the articles published in the field of cloud scheduling during 2008 and 2022, as mentioned in Figures 1A–C. Among them, most of the publications were published in 2018.

TABLE 2 Academic database.

Source	URL of the search engines	No. of returned articles
Google Scholar	http://scholar.google.com	360
Web of Science	http://www.webofknowledge.com	81
ACM Library	http://www.acm.org	19
IEEE Xplore	http://ieeexplore.ieee.org	122
Scopus	http://www.scopus.com	39
Springer	http://www.springerlink.com	34
ScienceDirect	http://www.sciencedirect.com	67
Total		722

3.5 Keyword search

In the formulation of our first question (RQ1), we particularly outlined the importance of VM scheduling and the necessity to improve its mechanisms due to the high rise in data accumulation and resource utilization. Based on this perspective and the growing interest of the researchers in VM scheduling, we only included peer review journal articles and conference papers from the most relevant digital libraries, as shown in Table 2. However, since we assumed that researchers and practitioners frequently use journals to obtain knowledge and disseminate new findings, we rejected conference papers that were not from trustworthy sources.

3.6 Scope of the study

Based on the standards outlined in the study's procedure, the major studies were included. The 67 articles included in this study are further divided into two categories: those that specifically address the VM scheduling challenge and those that examine various problem-solving strategies.

The literature review will provide a solution as follows:

1. What is the present status of VM scheduling in cloud computing?
2. What are the various methods used in VM management?
3. What types of research are carried out in this area?
4. Why is VM scheduling important in the area of cloud computing?
5. What are the approaches prevalent to solving VM scheduling problems?
6. Which approach may be opted for in the current cloud computing scenario?
7. How and why do VM scheduling approaches impact the performance of resource management in cloud data centers?
8. What are the challenges in the design and devilment of VM scheduling techniques in cloud computing?

Here, it is important to mention that the foremost attention of this study is VM scheduling, its architecture, and the techniques

TABLE 3 Abbreviation and illustration.

Abbreviation	Illustration	Abbreviation	Illustration
SLA	Service-level-agreement	SRC-I/O	Share reclaiming and collective I/O
SLR	Systematic-literature-review	SVS	Synchronization aware VM scheduling
DC	Data center	HEFT	Heterogeneous earliest finish time
VM	Virtual machine	CDM	Common deployment model
PM	Physical machine	AD	Active directory
IaaS	Infrastructure as-a-Service	PD	Passive directory
API	Application program interface	KVM	Kernel-based Virtual Machine
QoS	Quality of Services	DVMS	Distributed virtual machine scheduler
PM	Physical machine	BFD	Breadth first depth
VMP	Virtual machine placement	BALA	Bandwidth-aware lago allocator
VMM	Virtual machine management	VSA	VM scheduling algorithm
DVFS	Dynamic voltage frequency scaling	GRANITE	Greedy based virtual machine scheduling algorithm
WAN	Wireless area network	DCN	Data center network
EC	Evolutionary computing	VMSAGE	VM Scheduling Algorithm based on Gravitational Effect
EASE	Energy Efficiency and Proportionality aware Scheduling	FEM	Fairness-aware VM Scheduling Method
SMP	Symmetric multiprocessing	BFH	Best fit heuristic
ACO	Ant colony optimization	FHA	Find host algorithm
EEVS	Energy efficient scheduling	UTC	Bat algorithm
vCPU	Virtual central processing unit	BPA	Bandwidth provisioning algorithm

(Continued)

TABLE 3 (Continued)

Abbreviation	Illustration	Abbreviation	Illustration
QAC	Quality assessment checklist	PSO	Particle swarm optimization
LCA	League championship algorithm	MCKP	Multiple choice knapsack problem
GA	Genetic algorithm	CGDPS	Cost greedy dynamic price scheduling
CMU	Cumulative machine uptime	ACOPS	Ant-Colony Optimization and Particle Swarm-Optimization
MST	Maximum sustainable throughput	TLBO	Teaching learning based optimization
ERTE	Time and Resource Efficiency Metric	FCFS	First come first serve
PABFD	Power-aware best fit decreasing	LAVMS	Lock-aware Virtual-Machine Scheduling
VBP-Norm	Vector-Bin Packing Norm-based-Greedy Algorithm	MCT	Minimum completion time
CS	Cuckoo search	MET	Minimum execution time
KH	Krill herd	CIDD	Cloud intrusion detection dataset
SA	Simulated annealing	UTC	Universal time coordinated
DT	Dynamic thresholds	SOS	Symbiotic organisms search
VHEST	Virtualized homogeneous earliest start time	AWS	Amazon web services
PVLOCK	Para virtual spinlocks	WOA	Whale optimization algorithm
CRTS	Composition real-time scheduling framework	NP	Non-probabilistic
EASA	Energy-aware scheduling algorithm	VMM	Virtual machine monitor
FC	Fog computing	IoT	Internet of things

used in the literature to solve the VM scheduling problem. Hence, we do not concentrate on the other underlying elements of cloud scheduling such as task or job scheduling, workload scheduling, and workflow scheduling. In addition, the study does not consider VM migration in most cases. In the forthcoming section, the VM management methods are explained, and the abbreviation used throughout the review is shown in Table 3 with its illustration.

4 Virtual machine management

Virtual machine management is a solution for VM scheduling in the data center, which enables us to create and deploy the virtual host or VM and allocate or de-allocate the VMs, mapping the VMs with the PMs to provide better QoS as per user demand. The VM can be managed by different methods to achieve optimal resource utilization and cost saving. VM management methods consolidate the VMs on the physical machines without considering heterogeneity, which is one of the main aspects of modern-day data centers. Since finding the system's heterogeneity is essential to achieving considerable performance and effective resource management, it must be accounted for in designing VM management schemes. Many studies have been done on management strategies in cloud data centers; however, there is a lot to be explored for the schemes that can improve the effectiveness of data center.

4.1 Classification of VM management method

In this section, we put forward the underlined methods for VM management and their possible classifications. According to the investigation of the surveyed literature in this study, the methods or techniques involved in VM management can be classified as VM Scheduling, VM Allocation, VM Placement, VM Migration, VM Consolidation, and VM Provisioning. Things to be noted here, these methods are often used interchangeably in the literature, and the distinction between the actual methods used becomes challenging to identify. However, the main focus of this study is VM scheduling since an ill-managed VM scheduling on the data center in a heterogeneous environment not only leads to performance degradation of computing resources but also lowers energy efficiency, which results in more energy consumption (Sharifi et al., 2012).

This article focuses on VM scheduling since it has the following advantages: scalability, QoS, a particular environment, decreased overheads and latency, enhanced throughput, cost-effectiveness, and a more straightforward user interface. The VM management methods (see Figure 2) can be classified as below, whereas an overview of VM scheduling is shown in Figure 3.

- VM Scheduling: Allocating a group of VMs to a group of physical machines is the definition of a VM scheduling problem (Prajapati, 2013; Khan et al., 2018).
- VM Allocation: Allocating the user tasks to VMs is known as "VM allocation," and it often takes CPU, network, and storage requirements into account (Bouterse and Perros, 2017).
- VM Placement: It is a method for deciding which VM belongs to which physical machines (Chauhan et al., 2018).
- VM Migration: Relocating a VM means shifting it from one server or storage facility to another (Leelipushpam and Sharmila, 2013).
- VM Consolidation: As a result of the strategic placement of the VMs, we may reduce the number of necessary PMs (Corradi et al., 2014).
- VM Provisioning: Configurable actions linked to deploying and personalizing VMs following organizational needs (Patel and Sarje, 2012).

4.2 Systems model of VM scheduling

Figure 3 demonstrates the association between VMs and PMs. A sequence of all the PMs in the system here is represented as; $\rho = \{\rho_1, \rho_2, \ldots, \rho_{\aleph}\}$, $\aleph$ is the number of PMs, $\rho_i (1 \leq i \leq \aleph)$ which represents PM i. Whereas, VMs set on the PM $\rho_i \vartheta_i = \{\vartheta_{i1}, \vartheta_{i2}, ..., \vartheta_{im_i}\}$ in which m_i is the number of assigned VMs on PM i. Considering $S = \{S_1, S_2, \ldots, S_{\aleph}\}$ is the solution set which can be generated after the deployment of the VM ϑ on each physical machine. Hence, S_i is the resultant solution set when VM ϑ is mapped to PM ρ_i.

4.2.1 The formulation of the load

A workload of a PM generally can be derived by summing up the workloads of the VMs executing on it. We presume the finest time examined by previous data is τ, which is the period of τ from the existing time in the monitoring zone by previous data. According to the changing policies of PM workload, we can distribute the time τ into n times. Therefore, we define $\tau = [(t_1 - t_0), (t_2 - t_1), \ldots, (t_n - t_{n-1})]$. The equation states that, according to the changing policies of PM workload, the time τ is distributed into n smaller time intervals. In this notation, $(t_1, t_2, \ldots, t_{n-1})$ represent the end points of the n time intervals, and t_0 is the starting point. The values in the brackets represent the duration of each time interval, which is calculated as the difference between consecutive end points $(t_i - t_{i-1})$. The sum of all the duration of the time intervals is equal to the total duration of the period τ.

In the explanation $(t_k - t_{k-1})$, signifies time k. Assuming the workload of VMs is fairly constant every time, then we can define the workload of VM number in period k is ϑ (i, k). Thus, we can determine that in cycle τ, where n is the number of instances in the index I and workload(i) is the workload value for the ith instance. The workload of the VM ϑ_i on PM ρ_i is calculated by Equation 1:

$$\overline{\vartheta_i(i,\tau)} = \frac{1}{\tau} \sum_{k=1}^{n} \vartheta\,(i,k)\ \times (t_k - t_{k-1}) \tag{1}$$

Going by the system policy, the workload of a PM is generally derived by summing up workloads of the VMs executing on it as

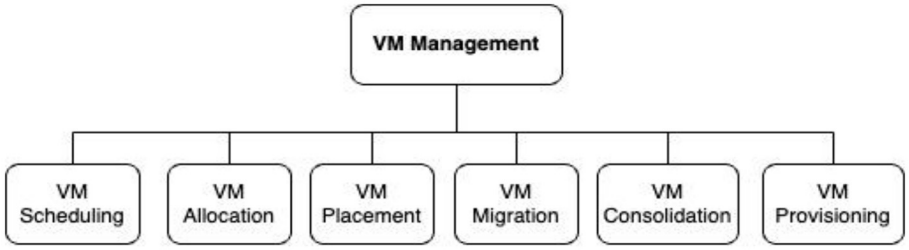

FIGURE 2
Classification of VM management techniques.

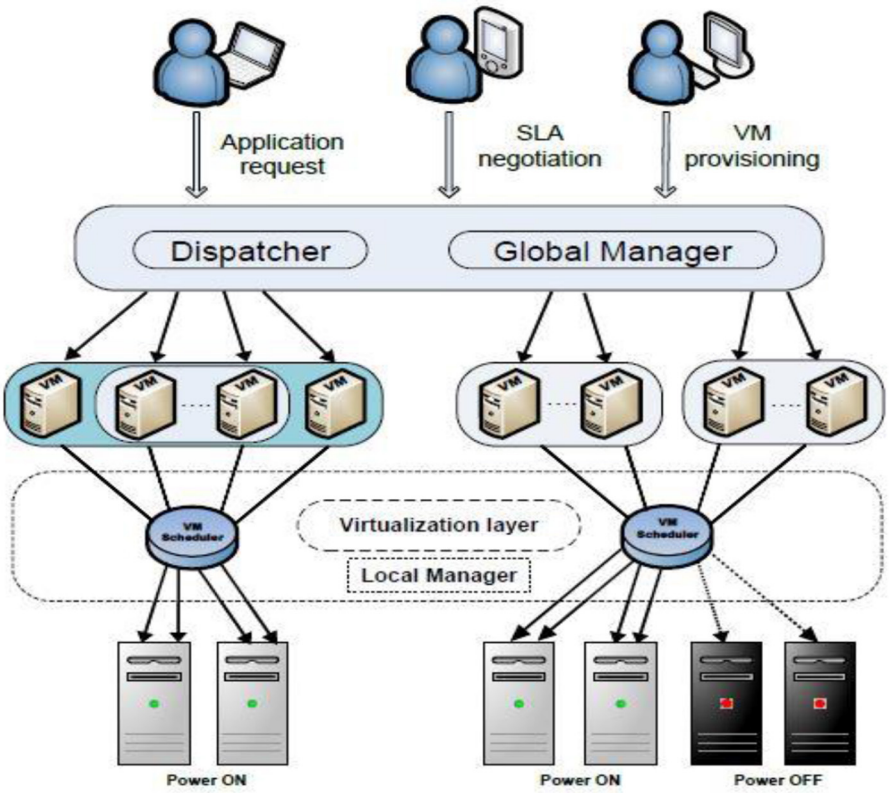

FIGURE 3
Virtual machine scheduling overview (Rana and Abd Latiff, 2018).

shown in Equation 2. Hence, we can assume that the workload of the PM p_i where mi is the number of VMs on PMj.

$$\rho\,(i,\tau) = \sum_{j=1}^{m_i} \overline{\vartheta_i(j,\tau)} \tag{2}$$

The present VM requires placement as ϑ. Then, the previous VM configuration is required by the current schedular, and the estimation of the workload of the VM is ϑ' based on historical data. Therefore, when ϑ is mapped to PM, the workload of each PM should be measured by Equation 3.

$$\rho\,(i,\tau)' = \begin{cases} \rho\,(i,\tau) + \vartheta' & \text{After deploy } \vartheta \\ \rho\,(i,\tau) & \textit{Others} \end{cases} \tag{3}$$

Typically, when ϑ is allocated to ρ_i, there will be some variations in the system workload. Consequently, to achieve load balancing, we must do load adjustments. The load discrepancy of the mapping solution S_i in time τ after ϑ is arranged to ρ_i by Equations 4 and 5.

$$\sigma_i\,(\tau) = \sqrt{\frac{1}{\aleph}\sum_{i=1}^{\aleph} \overline{(\rho(\tau)'} - \rho(i,\tau)')^2} \tag{4}$$

where

$$\overline{\rho(\tau)'} = \frac{1}{\aleph}\sum_{i\,=\,1}^{\aleph} \rho(i,\tau)' \tag{5}$$

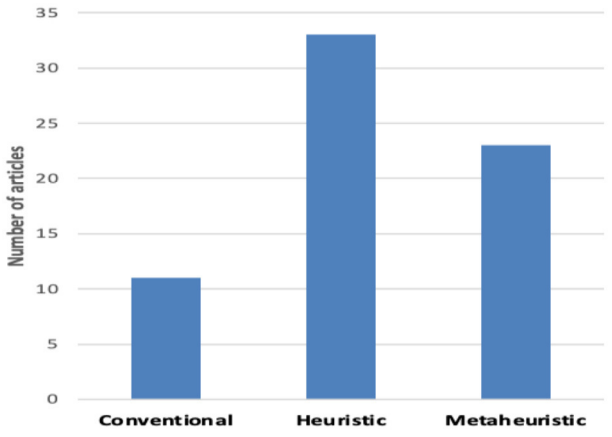

FIGURE 4
Distribution of articles based on the approach used in VM scheduling.

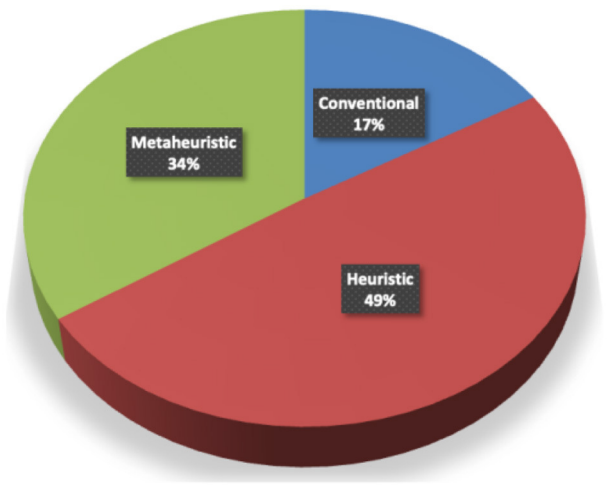

FIGURE 5
Percentage-wise distribution of approaches used in VM scheduling.

5 VM scheduling approaches

This review methodically segments the body of research on VM scheduling into three distinct methodologies. Initially, the traditional approach is detailed, highlighting fundamental scheduling techniques. The subsequent section digs into heuristic methods, which tailor problem-specific heuristic strategies for optimization challenges. The final section examines metaheuristic methodologies, embracing advanced, intelligent algorithms to tackle intricate VM scheduling in cloud computing. The objective of this study is to put forward the different methodologies tackling the same objectives from unique angles. Figures 4, 5 illustrate the distribution of these methodologies, while Tables 4–6 provide an exhaustive examination of the literature pertaining to each method, which is thoroughly discussed herein.

5.1 Conventional approach

Efficient VM scheduling techniques are proven to be efficient in solving problems, such as high response time taken by tasks, distribution of the VMs on the physical hosts to achieve optimal load balancing, equal resource consumption, and server consolidation in data centers. The mentioned problems are addressed using Best-Fit and Worst-Fit algorithms, which follow two mechanisms. The reaction time is reduced by a factor of ($\log_n$) for the best-fit method and by a factor of ($\log_n$) for the worst-fit method (1). In the worst-fit technique, the load on the PMs is equally distributed, but it requires additional VMs, such that every single host has to execute the processes. Then, in the best-fit process, every physical machine has equal resources left out for the execution of the remaining tasks. Better response times and more evenly distributed workloads on VMs are what the simulations suggest is possible. However, in the mentioned scheduling technique, they did not consider VM migration for the underutilized or overutilized host (Rahimikhanghah et al., 2022).

The study elaborates the distinction between VM scheduling and processor task scheduling in a traditional computing environment. In addition, it points out some key advantages and challenges of VM scheduling. The proposed gang scheduling-based co-scheduling algorithm works in two fashions. First, the algorithm schedules the coherent processes to run simultaneously on different processors. At the same time, it maps the related virtual CPUs (vCPUs) to the real processors. The simulation results exhibit faster execution of processes that execute on VMs and display higher performance and avoid unnecessary VM blocks (Salimi et al., 2012).

Hu et al. (2008) presented a novel scheme for VM scheduling using live migration of VMs to the under-loaded server clusters. The scheme named Magnet shows a better reduction in energy saving and is applied to both homogeneous and heterogeneous physical machines in the data center. The scheme also claimed an appositive impact on average job slowdown and a negative impact on the execution time for task processing. The authors of the study mentioned in the reference (Xia et al., 2008) measured the performance of interactive desktops and tried to solve the latency peak problem that arises during server peak workload. The proposed method enhances the XEN credit scheduler to analyze the latency for peak operation. They claim to reduce latency and frequency by their scheduler in comparison to the default one.

Von Laszewski et al. (2009) anticipated the Dynamic Voltage Frequency Scaling (DVFS) technique to analyze the problem of energy consumption in computer clusters. The proposed design focuses on the allocation of VMs on the DVFS-enabled clusters. The simulation results show an acceptable reduction in energy consumption. Lago et al. (2017) presented an optimization algorithm for VM scheduling considering bandwidth constraints in a heterogeneous network environment. These techniques work in two steps, first they used Find Host Algorithm (FHA) to find the optimum host to allocate the available VM which is executed by the cloud broker. Second, the Bandwidth Provisioning Algorithm (BPA) is used to provision the network bandwidth for the VM which is to be run on the host machine. In the simulation results, the proposed algorithm showed significant reduction in energy saving and a better makespan.

TABLE 4 Analysis of Conventional approach used in virtual machine scheduling.

References	Problem addressed	Algorithm /technique	Improvement/ achievement	Weakness /limitation	Tool /hypervisor
(Knauth and Fetzer, 2012)	VM Scheduling	OptSched Technique	Improve energy saving Reduce machine uptime	Does not work on real cloud Low resource utilization	Python
Pegkas et al. (2018)	VM Scheduling	Credit based algorithm	Improve response time Minimize the finish time	Low performance in all cases	Python
Takouna et al. (2011)	VM Scheduling	VM scheduling policy	High energy saving	Used basic DVFS mechanism Heterogeneous VM	Xen hypervisor
Imai et al. (2018)	Elastic VM Schedling technicus	New framework for proactive elastic VM scheduling	Better QoS	Single objective application Low scalability	Not mentioned
Bazarbayev et al. (2013)	VM Scheduling and Placement	Content based scheduling algorithm	Improve network utilization Reduce network congestion	High response time	Ubuntu Server
Rathore and Chana (2014)	VM-Placement technique, Load Balancing, Server Consolidation	Best-fit and Worst-fit algorithm	Reduce response time Better resource utilization Cost saving	Do not consider underutilized host and over utilized host for migration	CloudSim
Salimi et al. (2012)	Scheduling advantages and optimization	Virtual Processor co-scheduling method	Increase system performance	The work performs only 4 tasks	CloudSim
Lago et al. (2017)	VM Scheduling	Dynamic Voltage Frequency Scaling (DVFS)	Energy saving	Heterogeneous VM	CloudSim
Hu et al. (2008)	VM Scheduling	Magnet	Reduce Energy	High execution time	Xen hypervisor
Xia et al. (2009)	VM Scheduling	Not mentioned	Reduce latency	Basic approach adopted	Xen hypervisor
Von Laszewski et al. (2009)	VM Scheduling	Not mentioned	High energy saving Reduce CO_2 emission	Result showed discrepancy in data	nBench, Linux and DVFS-SIM /OpenNebula

In VM scheduling of heterogeneous multicore processor environment, two key issues are significant to achieve an efficient performance. Characteristics of VM for optimum VM placement at the suitable core and the actual source of delay to eliminate the impede cloud performance. The authors of the study mentioned in the Takouna et al. (2011) developed a plan to allocate resources among the several VMs. The authors discuss performance dependence on the physical host and responsiveness to CPU clock frequency. The simulation outcomes show that the proposed scheduling policy is effective in energy saving in a cloud environment. In a cloud data center, excessive amount of energy is consumed by the VM scheduler. Knauth and Fetzer (2012) suggested the energy-aware scheduling algorithm OptSched to minimize energy-saving problems in cloud computing. Simulation results show that the enhanced method can significantly reduce CMU up to 61.1% when compared with the default scheduler round-robin and is considered the best fit in OpenStack, OpenNebula, and Eucalyptus.

One other study proposes a credit-aware VM scheduling method to reduce data center overhead. The mechanism seems to be easy to implement with a simplified design. However, the experimental result does not show optimal performance in all cases and is not even implanted in the real cloud (Pegkas et al., 2018). In stream data processing, the demand of the workloads changes over a period of time. To maintain seamless processing, the VMs need to allocate and deallocate frequently by the VM Manager (VMM). In this so-called steam processing scenario, maintaining QoS is a challenging task and requires adaptive scheduling techniques to handle uncertainties. Imai, Patterson (Imai et al., 2018) provided a proactive elastic VM scheduling framework to forecast the arrival of workloads; when the estimation is done for the arrival of the highest workload, the minimum amount of VM is allocated to handle that workload. To know the uncertainties from VM and application, they have used MST (maximum sustainable throughput) model. The authors applied their framework on three different workloads and were able to achieve 98.62% of QoS satisfaction and 48% less cost in comparison to static scheduling.

On the other hand, there is a high possibility to discover a high amount of content similarity and identical disk blocks with a similar operating system and the same host with the help of VM scheduling. The researcher observed that a similarity between VM images can be as high as 60%−70% which causes a reduction in the amount of data transfer in the VM deployment process. Based on the above notion, Bazarbayev, Hiltunen (Bazarbayev et al., 2013) developed a content-based scheduling scheme to reduce the network congestion which is related to the VM disk image transfer process inside data centers. Data center network usage and congestion are significantly reduced as a result of the

algorithms' evaluation, which shows a reduction in data transfer of up to 70% during the processes of VM migration and virtual disk image transfer.

Conventional VM scheduling methods, including Best-Fit and Worst-Fit algorithms, have effectively addressed core issues within cloud computing environments, such as task response time optimization, equitable VM distribution for load balancing, and server consolidation. These techniques, grounded in simplicity, have proven to be quite robust, with Best-Fit methods reducing reaction times significantly and Worst-Fit approaches ensuring even workload distribution across physical machines. Additionally, innovations such as the gang scheduling-based co-scheduling algorithm have introduced improvements in running coherent processes simultaneously, thereby enhancing the overall execution performance.

However, these traditional methods are not without their limitations. They often do not account for the dynamic aspects of cloud computing, such as VM migration to balance the load on underutilized or overutilized hosts, which can be crucial for maintaining efficiency and responsiveness in data centers. While these methods have laid a strong foundation for VM scheduling, their lack of adaptability in rapidly changing environments and their potential for increased resource requirements highlight the need for more advanced approaches to address the evolving complexities of VM management.

5.2 Heuristic approach

The heuristic approach to handling complex optimization problems is explained as trying to find a probable number of solutions to an NP-hard problem and suggest the best solution to achieve some specific objective function. It is mostly bound with hard and soft constrains which must not be overlooked in the optimization design. Heuristic approaches perform where traditional approaches fail, especially in the high dimensional or multimodal space when a problem can be addressed using more than one solution. In this context, many researchers have applied heuristic approaches in their study and achieved effective solutions to their problems. Table 5 shows a descriptive analysis of the heuristic approach. We discuss here the heuristic approaches used to solve VM scheduling problems.

In the SMP (Symmetric Multiprocessing) VM scheduling, dynamic load balancing and CPU capping techniques are used, which consequently result in a significant number of inefficiencies in parallel workloads. In a virtualized system, where the tenants rent the resources, fairness among them considered being the key to success in running their applications effectively. However, the available virtualization platforms do not implement fairness in a condition where some VMs contain several virtual CPUs running on different CPUs. Based on this method, Rao and Zhou (2014) developed an innovative vCPU scheduling technique, namely, Flex, which applies fairness at the VM level and also increases the effectiveness of parallel running applications on the host servers.

In other progress, an efficient dynamic VM scheduling, the algorithm is developed to address the energy-consumption problem with the concentration of deadline constraints (Uddin et al., 2014). The study presents a robust energy-efficient scheduling technique, namely EEVS, which can be capable of dealing with various physical nodes and equally performs in a dynamic voltage environment. Furthermore, the algorithm considers scheduling periods and optimal performance-power ratio as performance parameters. Experiment analysis shows that in the best instances, VMs can reduce their energy consumption by over 20% while increasing their processing power by 8%.

According to Quang-Hung and Thoai (2015), Time and Resource Efficiency Metric (ERTE) is a suggested technique for scheduling VMs that take energy efficiency into account to reduce data center idle time. In addition, the suggested approach was evaluated in terms of power consumption alongside two state-of-the-art algorithms, namely, power-aware best fit decreasing (PABFD) and vector bin packing norm-based greedy algorithm (VBP-Norm L1/L2). Based on experimental results, the suggested scheduling method not only improves performance by 48% but also reduces average energy usage by 49%.

In the virtualized environment and with the presence of an intensive mixed workload, reducing energy consumption is considered one of the challenging tasks. According to Xiao et al. (2014), to reduce the energy consumption caused by I/O virtualization, a mixed-workload energy-aware VM scheduling technique was developed. Additionally, they developed a novel scheduler called SRC-I/O by fusing two newly designed techniques, namely, share-reclaiming and communal I/O. Both the share-reclaiming method and the collective I/O method aim to increase CPU utilization and reduce context-switching costs due to I/O-intensive workloads, respectively. Simulation results reveal that the SRC-I/O scheduler outperforms its rival on a different performance matrix.

Increases in virtualization technologies have allowed for massive VM consolidation in data centers. Services that depend on rapid responses could be hampered by a lack of availability if they did not have access to latency-sensitive task support. In this regard, Kim et al. (2009) accommodate latency-sensitive tasks, and it is necessary to devise a priority-based VM scheduling method that takes into account the needs of guests. The provided method schedules the required VMs for workload allocation based on the priority of the VMs and the current state of the guest-level tasks running on each VM. In addition, it selects for scheduling those VMs that are capable of running latency-sensitive applications with the quickest possible response to I/O events. The study mentioned in the Zhao et al. (2014) reduces the VM's carbon footprint by putting forward a cognitive scheduling method based on its camera's eye. The suggested method seeks to identify the optimal PM to allocate to a VM so that it may run within a specified response time. When compared with other algorithms, this one is 17% more efficient at saving power. Due to SLA violations of up to 14%, the proposed algorithm does not achieve optimal performance with response time.

Due to high flexibility and cost-effectiveness, multiple applications run concurrently on the virtual cloud. Running tightly-coupled parallel applications is a feasible solution over the clustered cloud environment for better resource utilization. However, due to over-commitment in the cloud and ignorance of the synchronization constraint of VMs by Virtual Machine

TABLE 5 Analysis of Heuristic approach used in virtual machine scheduling.

References	Problem addressed	Algorithm /technique	Improvement /achievement	Weakness /limitation	Tool /hypervisor
Rao and Zhou (2014)	Dynamic VM scheduling	Symmetric Multiprocessing (SMP) based VM scheduling scheme (Flex)	• Achieve fair CPU allocation	• Low performance	Xen 4.0.2
Ding et al. (2020)	VM scheduling	Energy efficient VM scheduling (EEVS)	• Reduce energy consumption • Low execution time	• Power penalties of status transitions • VM migrations are ignored	Not mentioned
Quang-Hung and Thoai (2015)	VM scheduling	ETRE algorithm	• Low total busy time of all PMs	• Does not consider other parameters	CloudSim
Xiao et al. (2014)	VM scheduling	Shared reclaiming with collective IO scheduler (SRC-I/O)	• Minimum CPU utilization • Better energy efficiency	• Low scalability	Xen Hypervisor
Beloglazov and Buyya (2012)	Dynamic VM scheduling	Dynamic Thresholds (DT)	• Improve energy consumption • High level of SLA	• Applied on single core CPU only	CloudSim
Kim et al. (2009)	VM scheduling	Priority-based scheduling scheme	• High timeliness and CPU fairness • Low response time	• Required kernel modification to implement the scheme • Used open-source OS that may encounter security issues	Xen 3.0.4, para-virtualized Linux 2.6.16
Zhao et al. (2014)	VM Scheduling	Vision cognition algorithm	• Improve energy saving	• SLA violation	CloudSim
Wu et al. (2018)	VM scheduling	Synchronization aware VM scheduling algorithm (SVS)	• High application performance • Better execution time	• Low resource utilization	Xen hypervisor
Ebrahimirad et al. (2015)	VM scheduling	Virtualized homogeneous earliest start time (VHEST)	• Improved utilization • Makespan reduction • Reduced power consumption	• Homogeneous VMs	VDCS (Virtualized data center simulator)
Saravanakumar and Arun (2016)	VM scheduling and VM placement	Common deployment model (CDM)	• High resource utilization	• Compared with iCanCloud	CloudSim
Xie et al. (2014)	VM scheduling scheme	Process-aware predictive scheduling	• Improve disk I/O speed of the process	• Based on only Xen hypervisor	Xen Hypervisor
Kim et al. (2009)	VM scheduling scheme	Task aware VM scheduling scheme	• High performance	• One vCPU on single CPU • Does not consider task migration and synchronization issues	Xen Hypervisor
Miao and Chen (2015)	VM scheduling scheme	FlexCore scheduling scheme	• High performance	• Does not consider VM migration	KVM Hypervisor
Kertesz et al. (2016)	VM scheduling	Pliant-based VM scheduling	• Low execution time	• Cost saving for provider's only	CloudSim
Quesnel et al. (2013)	VM scheduler	Distributed VM scheduler (DVMS)	• High system utilization	• Does not show the QoS improvement	KVM hypervisor
Adhikary et al. (2013)	VM scheduling	VM scheduling algorithm (VSA)	• Better energy conservation	• Worked for network devices with fixed experiment condition	CloudSim
Seo et al. (2014)	VM Scheduler	Composition real-time scheduling framework (CRTS)	• Low power consumption	• Worked with only two VMs	RT –Xen Hypervisor
Li X. et al. (2018)	VM Scheduling	Greedy based holistic approach (GRANITE)	• Reduce Energy consumption • Low SLA violation	• Do not compare with benchmark systems	CloudSim

(Continued)

TABLE 5 (Continued)

References	Problem addressed	Algorithm /technique	Improvement /achievement	Weakness /limitation	Tool /hypervisor
Wu et al. (2018)	VM Scheduling	Maximum elasticity scheduling	• High computation time • High communication elasticity	• Feasible for cloud data center network (DCN)	Matlab
Li et al. (2013)	VM Scheduling	Hierarchical VM placement algorithm	• High resource utilization	• Homogeneous VMs	Matlab
Xu H. et al. (2018)	VM Scheduling	Gravitational effect based VM scheduling (VMSAGE)	• Reduce energy consumption • Minimize migration time	• Compare with on conventional BFS and DVFS	CloudSim
Xu H. et al. (2018)	VM Scheduling	HSM scheduling method	• Better load balancing • Improve resource utilization	• Compare with on conventional FFD and BFD	CloudSim
Lago et al. (2017)	VM Scheduling	Bandwidth-aware lago allocator (BALA)	• Low energy consumption • Low makespan	• Performance degradation	CloudSim
Al-Dulaimy et al. (2018)	VM Scheduling	Multiple choice knapsack problem (MCKP)	• Reduce energy consumption • Improve PMs utilization	• Live migration cost overhead	CloudSim
Xu et al. (2019)	VM scheduling	Cost-greedy dynamic price scheduling algorithm (CGDPS)	• Enhance execution time • Improve cost saving • Increase fairness of users	• CPU cores cannot be allocated to more than one VM	Not mentioned
Yu et al. (2019)	VM placement	Lock-aware VM scheduling scheme (LAVMS)	• Reduce CPU waiting time • High performance	• Implemented on limited VM	Xen-based prototype
Qiu et al. (2019)	VM scheduling	Energy efficiency and proportionality aware scheduling (EASE)	• Low energy consumption • High completion time	• Work for few numbers of VMs	KVM/QEMU
Xu H. et al. (2018)	VM scheduling	VM scheduling heuristics	• Reduce energy consumption • High performance	• Implemented does not support VM migration	CloudSim
Xing et al. (2017)	VM scheduling	Fairness-aware VM scheduling method (FEM)	• Improve fairness • High power saving	• Low resource utilization • Low scalability	CloudSim
Xu et al. (2012)	VM scheduling for WMAN	MFEA Scheduling technique	• Energy saving	• Resource wastage	Not mentioned
Wan et al. (2020)	System queuing scheduling model	Particle optimization	• Performance and Cost	• Semi Metaheuristic approach	Matlab
Qi et al. (2020)	VM Scheduling	QVMS using NSGA-III	• Energy and downtime	• Increased migration cost	NA
Saravanakumar et al. (2021)	Clustering based VM scheduling	Cloud radio access network (C-RAN)	• Network-overhead, allocation time	• Data size volume constraints ignored, Work in only homogeneous environment	CloudSim
Xu et al. (2023)	VM Scheduling	Greedy-based best fit decreasing (GBFD)	• QoS	• Dynamic workload overlooked	CloudSim

Monitor (VMM), performance degradation is taken into consideration in recent research. To overcome this problem Wu et al. (2018) emphasized the role of dynamic workload on the VM in a Data Center Network (DCN) and presented a VM scheduling to improve the elasticity as a new QoS parameter. A new precedence-constrained parallel VM consolidation algorithm is anticipated by the study mentioned in the reference (Ebrahimirad et al., 2015), which tends to improve the resource utilization level of physical machines and also displays minimum energy consumption. Simulation results show that their algorithm

performs better in comparison to Heterogeneous Earliest Finish Time (HEFT) in reducing energy and make span time of the services.

Based on a brokering mechanism, Saravanakumar and Arun (2016) proposed a Common Deployment Model (CDM) to manage VM in cloud data centers efficiently. After a task has been completed, the current state of the VM is preserved using the active directory (AD) and passive directory (PD). These folders are used for two processes, VM migration and VM rollback, ensuring that VMs have the correct configuration mapping of the physical computers. The suggested model takes into account VM downtime for various job kinds. When it comes to managing unused VMs in a repository, the CDM model is contrasted with the iCanCloud concept. Keeping the inactive VM in the hypervisor eliminates the latency issue that arises when moving VMs between the hypervisor and the VM repository. The experimental results show that the CDM-based model takes less latency in VM management. They proposed two algorithms for VM scheduling and VM placement to achieve effective utilization of VM. Furthermore, they have compared both algorithms with different scheduling and placement algorithms, respectively. VM scheduling algorithms show a better result when compared with other algorithms regarding CPU utilization, whereas VM placement resulted in better improvement in terms of completion time of VM placement and resource utilization.

I/O performance degradation is a common phenomenon in a virtualized environment. The VM is not able to distinguish the different processes coming from the same physical machine. Since the process information is located in the higher layers, getting it can be challenging. To address this problem, Xie et al. (2014) suggested a disk predictive scheduling method that takes into account running processes be used to solve the disk I/O issue. With the assistance of a predictive model, the VMM in this approach learns about the process and then uses that knowledge to categorize the I/O request. The connection between a process and its address space is used to infer the level of awareness of the process. The simulation results validate the practicability of the proposed strategy and highlight the subsequent increase in disk I/O speed.

In a multi-core virtualized environment, Symmetric Multiprocessing (SMP) is increasingly being used for efficient resource utilization and performance degradation. A separate scheduler exists in the hypervisor and in the guest host, resulting in a problem of double scheduling. To overcome this problem, Miao and Chen (2015) evaluated a scheduling scheme FlexCore using vCPU ballooning. The scheme dynamically adjusts the number of vCPUs of a VM at runtime and eliminates unnecessary scheduling within the hypervisor layer to considerably improve the performance. The experimentation is done on a KVM-based hypervisor which shows that the average performance improvement is approximately 52.9%, ranging from 35.4% to 79.6% for a 12-core Intel machine for PARSEC applications. In a similar progress, Kertesz et al. (2016) presented an improved pliant-based VM scheduling scheme for solving energy consumption problems. The authors in their study utilized industrial application workloads to evaluate the performance of their improved CloudSim framework. The results depict a significant improvement in energy saving and a better trade-off in execution time.

Due to the hard scalability problem in a distributed virtualized cloud environment, it is difficult to manage VMs by VM In-charge on a pool of physical machines. It becomes worse in the case of VM image transfer. In this regard, Quesnel et al. (2013) provided a new Distributed Virtual Machine Scheduler (DVMS), which acts as a decentralized and preemptive scheduler in a massive-scale distributed environment. As shown in the results, the elements of the validation approach are sufficiently solving the resource violation problem.

In another type of progress, Adhikary et al. (2013) suggested a distributed and localized VM scheduling algorithm (VSA) to cater to energy consumption problems in data centers. The proposed algorithm functions as intra-cluster and inter-cluster scheduling and addressed some major parameters, such as energy, resource estimation, and availability. It schedules VMs in a way that energy consumption is minimized for both servers and networking devices. The results show that the algorithm outperforms other existing algorithms in terms of energy reduction.

VM consolidation is often used to solve energy consumption problems. Second, energy consumption can also be managed by sending the real-time resource requirement to the VMM and controlling the frequency of recourse demand. In that essence, a power-aware framework is introduced for compositional real-time scheduling. The method encapsulates each VM into a single component to minimize resource utilization and thus reduce energy. The framework is implemented on Xen hypervisor on Linux kernel, resulting in better performance (Hu et al., 2010).

Efficient VM scheduling increases the performance of the data center and increases the profitability of the cloud providers. In this regard, Li X. et al. (2017) offered a greedy-based VM scheduling algorithm GRANITE to reduce datacenter energy consumption following two major strategies, namely, VM placement and VM migration. They have used computational fluid dynamics techniques to address the cooling model of the datacenter. Moreover, they claim to address the CPU temperature for the first time along with the other infrastructure devices and nods. The results show that the algorithm outperforms other existing algorithms in terms of energy reduction. In a different study, Li W. et al. (2017) improved the deficiency of the semi-homogeneous tree to a general heterogeneous tree as its optimal solution. Using a hose model, the proposed maximum elasticity scheduling optimizes both maximum elasticity computation and maximum elasticity communication. Inspired by the gravitational model of physics, Xu et al. (2019) presented a VM Scheduling Algorithm based on Gravitational Effect (VMSAGE) to handle the issue of energy consumption in data centers. This study is the extension of the study mentioned in the Rahbari (2022) in which the authors presented a heuristic-based approach for VM scheduling for Fog-cloud. To assure optimum utilization of the resources, their method addressed the issue of load balancing and achieved better resource utilization on the edge network.

Using Virtual Machine Management (VMM) strategy, Al-Dulaimy et al. (2018) anticipated an improved energy-efficient VM scheduling technique for dynamic consolidation and placement of

the VMs in data centers. In this strategy, Multiple Choice Knapsack Problem (MCKP) first decides the set of VMs to migrate from the under loaded and overloaded PM criteria. Then, VM selection is performed from the generated candidate solutions, and finally, this selected VM is placed on the number of PMs. The proposed method outperforms when compared with similar strategies in terms of energy saving.

In a similar study, Xu H. et al. (2018) investigated the VM scheduling problem and proposed an incentive-aware scheduling technique for both cloud providers and cloud users with a guaranteed QoS. In this study, the improved meta-heuristic method, namely, Cost Greedy Dynamic Price Scheduling (CGDPS) prioritizes the VM requests as per the user demand and generates several candidate solutions. Finally, the VMs are assigned to the candidate node with minimum computation cost. The comparative results show a competitive improvement in user satisfaction.

In the study by Yu et al. (2019), a synchronization problem in VM scheduling is addressed to avoid the extra-long waiting time assigned to a vCPU for lock spins. The proposed Lock-aware Virtual Machine Scheduling (LAVMS) provides additional scheduling chances for processors to avoid locks. The method ensures the scheduling without wasting the waiting time of the vCPU. The scheme outperforms when compare with the contemporary para-virtual-spinlocks (PVLOCK) in terms of performance. Along the same lines, Qiu et al. (2019) introduced an energy efficiency and proportionality-aware VM Scheduling framework (EASE). The framework set out the standard benchmarking as per the specified configuration components of the servers. Again, it addresses the real workload which again configuration-centric to the servers. Then, real-time server data are collected, efficiency is identified, and finally, workload classification is performed to achieve optimum VM scheduling. The simulation results depict a significant reduction in energy and completion time up to 49.98% and 8.49%, respectively, in a homogenous cluster. Similarly, in heterogeneous clusters, the observed reductions in energy and completion time are 44.22% and 53.80% respectively.

Considering resource provisioning a major concern for IoT applications, the study mentioned in the Xing et al. (2017) adapted a fairness-aware VM scheduling method (FEM) to achieve fairness and energy saving. Therefore, the system is designed and evaluated on three IoT datasets and compared with the benchmark energy-efficient VM scheduling (EVS). The experimented graphs show superior performance in resource-fairness and power saving. In the same context, Xu et al. (2020) considered the balancing scenario between energy saving with guaranteed performance and introduced a novel VM scheduling technique for Cyber-physical system. The joint-optimization model-based method utilizes the live migration of the VMs to underloade PMs and offload the overhead, consequently reducing power consumption and performance degradation. The study mentioned in the Xu H. et al. (2018) examined the power management problem in Wireless Metropolitan Area Network (WMAN) and put forward a VM mapping strategy to reduce power consumption. The proposed method, namely, MFEA, is optimized to reduce the number of VMs on the physical servers after migrating the underutilized VMs. The experimental graph shows comparable energy reduction with other benchmark techniques.

In a different study Wan et al. (2020) offered a system queuing scheduling model to analyze the performance of the cloud systems by switching off and on the (hot and cold shutdown) VMs. The proposed method uses multi-objective particle optimization to optimize the most critical parameters in the cloud scheduling process, such as performance and cost. However, the heuristics approach is not used in the true sense, and the description is lacking. Similarly, Qi et al. (2020) developed a QoS-aware cloud scheduling system by applying the NSGA-III algorithm to find the optimal VMs to migrate on the PMs in the cyber-physical system (CPS). The algorithms generate multiple VM scheduling solutions and select the best strategy to map the VMs. In another study, Saravanakumar et al. (2021) proposed a VM clustering method to monitor the performance measure of the VM metrics, such as network-overhead cost. It dynamically allocates the submitted tasks to the VMs to deal with the network overload problem and reduce the allocation time. However, the proposed method lacks in dealing with the volume of the data size constraints. Furthermore, Xu et al. (2023) addressed one of the significant factors called reliabilities in VM scheduling and presented a fault tolerance scheduling system while satisfying several QoS. They designed a greedy-based technique to identify suitable computer nods to execute the user's tasks with improved performance.

The heuristic approach to VM scheduling has been instrumental in addressing the challenges of resource management in cloud computing environments. Methods such as Flex, which prioritize fairness and EEVS, designed for energy efficiency, demonstrate the versatility and effectiveness of heuristic strategies. These approaches consider a variety of performance parameters, such as load balancing, energy consumption, and CPU utilization to improve the overall performance of cloud services. They excel particularly in scenarios requiring dynamic load balancing and in systems where resources are rented, ensuring fairness and efficient application performance.

However, heuristic methods are not without drawbacks. For instance, some may not fully support the complexities of VM migration or may not effectively address the latency issues during peak server workloads. Moreover, while these methods aim to reduce energy consumption and improve resource utilization, they sometimes fall short in terms of scalability and in addressing the synchronization constraints of VMs in cloud environments.

In conclusion, while heuristic methods for VM scheduling have shown a better response to the dynamic nature of cloud computing and have provided solutions to specific optimization problems, they still face challenges. Improvements are needed to enhance their adaptability to rapid changes in workload and better address the intricacies of VM migration and consolidation. Despite these challenges, heuristic approaches remain a critical component of the VM scheduling toolkit, offering a range of solutions that traditional methods cannot provide.

5.3 Meta-heuristic approach

The distinction between heuristic and meta-heuristic is overwhelming. Both heuristic and meta-heuristic approaches are

TABLE 6 Analysis of Meta-heuristic approach used in virtual machine scheduling.

References	Problem addressed	Algorithm /Technique	Improvement /Achievement	Weakness /Limitation	Tool /Hypervisor
Hu et al. (2010)	Load balancing	Genetic algorithm (GA)	• Reduce load imbalance • Low migration cost	• High makespan	OpenNebula and C++
Kumar and Raza (2015)	VM Scheduling and Placement	Particle swarm optimization-based policy	• Reduce resource wastage	• Low performance • High server utilization	Eclipse Kepler 2
Cho et al. (2015)	VM Scheduling	ACO-based VM scheduling (ACOPS)	• Improve resource utilization	• Work on single objective • Homogeneous synthetic cloud	Test-bed@NCKUEE
Gondhi and Sharma (2015)	VM Allocation	Local search-based Ant colony optimization	• Reduce energy consumption • Better resource utilization	• Only one optimal solution • Compared only with BFD	CloudSim
Liu et al. (2017)	VM Scheduling	Adaptive penalty function (CGA)	• Improve deadline constraint	• Independent task • Save execution cost	WorkflowSim
Wang et al. (2018)	VM Scheduling	Improved teaching learning-based optimization scheduling strategy (TLBO)	• High energy saving	• Does not compare with benchmark algorithms	Not mentioned
Qin et al. (2019)	VM Scheduling strategy	Semi sleep mode VM scheduling	• High energy saving • Improve average latency	• Does applied on real time workload • No comparison shown	Matlab 2010a
Xu and Li (2019)	VM Scheduling methods	Learning effects models	• High execution time • Reduce makespan	• Work for single VM only • Does not show practical implementation	MapReduce
Zhao et al. (2014)	VM placement	Divide and conquer strategy with branch and bound algorithm (DCBB)	• Low execution time • Better convergence speed	• Yet to prove theoretically • Algorithm adaptation on DCBB is not clear	Amazon Elastic Compute Cloud (EC2)
Sui et al. (2019)	VM Scheduling	Genetic algorithm based SVR_GA for classification, Differential evaluation based adaptive algorithm for local search (ESA_DE)	• Reduce energy • Low virtual migration	• Low scalability • Increase throughput	CloudSim
Li Y. et al. (2018)	Dynamic VM scheduling	GA based dynamic VM scheduling strategy	• Improve utilization • Better load balancing	• No significant results	CloudSim/OpenStack
Feng and Zhu (2019)	Predictive VM Scheduling	Revivification-based prediction (ERP) model and ERPA	• Reduced execution time	• Conservative time synchronization schema	Java
Karthikeyan and Soni (2020)	VM Scheduling	GA, variable neighborhood search (VNS) and PSO based approach	• Utilization and Completion time	• Did not mentioned algorithm improvement	CloudSim
Kruekaew and Kimpan (2020)	VM Scheduling	Enhanced ABC	• Makespan and degree of imbalance	• High recourse cost	Matlab
Naik et al. (2020)	VM migration	Fruit fly Hybridized Cuckoo Search (FHCS)	• Energy and resource leakage	• Did not considered deadline constraints	CloudSim
Rana et al. (2022)	VM Scheduling	M-WODE	• Makespan and Cost	• Migration cost ignored	CloudSim
Medara and Singh (2021)	VM Scheduling	EASVMC	• Energy reduction and utilization	• Deadline constraint ignored	WorkflowSim
Ajmera and Tewari (2021)	VM Scheduling	VMS-MCSA	• Energy	• Tested on synthetic workload	CloudSim
Chaudhury (2021)	VM Scheduling	Particle Swarm optimization and Ant Colony Optimization approaches called (PSACO).	• Load Balancing, energy	• High computational cost	CloudSim

(Continued)

TABLE 6 (Continued)

References	Problem addressed	Algorithm /Technique	Improvement /Achievement	Weakness /Limitation	Tool /Hypervisor
Alsadie (2021)	VM Scheduling	Metaheuristic framework called MDVM	• Energy usage, makespan and cost	• High computational cost, homogeneous environment considered only	CloudSim
Ss and Hs (2022)	VM Scheduling	GA based Technique	• Energy usage, utilization	• SLA Violation in VM migration	CloudSim
Sheng (2022)	ML based VM scheduling prediction system	SchedRL	• Allocation time	• Increased computational time	Python

used to solve high-dimensional and multi-model problems and provide near to optimal solutions for a problem. Heuristic approaches are problem-specific, whereas meta-heuristic approaches are more generalizable and adaptable. The latter can guide, modify, and hybridize with other heuristic approaches in the process of local optima generation (Mukherjee and Ray, 2006).

Nature-inspired meta-heuristics contain immense power in solving complex engineering problems. Meta-heuristic approaches have unique features in striking a balance between exploration and exploitation phases and avoiding local optima stagnation (Ss and Hs, 2022). Due to these unique and promising features, researchers across the world prefer using meta-heuristic approaches in their efforts to solve optimization problems. In this section, we discuss the most relevant metaheuristic approaches used in solving VM solving problems. Table 6 shows a brief analysis of the meta-heuristic approach. Furthermore, the parameters used in this surveyed literature are shown in Table 7. In Figures 6, 7, the distribution of the literature based on parameters used in numbers and percentages is mentioned. Figures 8, 9 show a comparison between single-objective and multi-objective optimization problems used in the literature. Table 8 maintains a list of available datasets for cloud computing. Here, in this subsection, we talk about using various meta-heuristic techniques to address VM scheduling issues.

In a cloud environment that has been virtualized, the incoming requests frequently change. The types of requests a VM may get and the tasks it will carry out are unknown to the system. Therefore, a technique either considers a fixed number of tasks or requires detailed information about the tasks, which has become insignificant. In this regard, Cho et al. (2015) introduced a hybrid meta-heuristic approach that incorporates ACO and PSO, two highly developed algorithms, to tackle the VM scheduling problem. To anticipate incoming workload and adapt to changeable settings, the proposed ACOPS algorithm employs previously stored information on the server. To save computing time, it does not require any more job information and disproves unmet scheduling needs. The simulation graphs demonstrate that the suggested algorithms outperform other comparable systems and have a balanced cognitive burden. In a cloud environment that has been virtualized, the incoming requests frequently change. The types of requests a VM may get and the tasks it will carry out are unknown to the system. Gondhi and Sharma (2015) developed a VM allocation problem solution based on the ACO algorithm. The authors modified the ACO by using a local search algorithm to maximize the allocation result because they believed that the combinatorial problem of bin packing was NP-hard.

VM scheduling can be perceived as the allocation and placement of several VMs to a set of PMs. In this regard, Kumar and Raza (2015) proposed an enhanced VM scheduling policy for VM allocation in cloud data centers based on particle swarm optimization (PSO). The suggested policy intelligently distributes the VMs among the fewest possible physical hosts, reducing resource costs. According to the findings, the strategy not only reduces the number of VMs allocated to the host machines but also improves performance and scalability.

There are common pitfalls in existing evolutionary algorithms, such as defining problem-specific parameters for constrained optimization problems and their static nature, which can lead to premature crossover. Liu et al. (2017) provide a metaheuristic approach using an adaptive penalty function for workflow scheduling to enhance time constraints. When compared with existing state-of-the-art algorithms, the presented algorithms perform admirably and produce reasonable results under constraints, such as time and money.

In another progress, Zhou and Yao (2017) developed a revolutionary scheduling method based on teaching and learning optimization (TLBO) to cut down on energy use. It divides the VM scheduling into two, one pool of the VMs is to keep in active mode to cater for the arrival of a dynamic workload. The second pool of VMs is kept in reserve and put in low energy saving mode or sleep mode. The reserve pool of VMs allocated and deallocated based on resource demand. In a different study, the authors of the study mentioned in the reference (Rana and Abd Latiff, 2018) presented a whale optimization algorithm (WOA)-based cloud framework for multi-objective VM scheduling in data centers.

Qin et al. (2019) proposed a semi-sleep mode issue in VM scheduling, which was considered, and a plan to decrease the average latency of resource requests, which was offered to help preserve power in data centers. In their proposed system, the authors introduced a cost function to optimize the semi-sleep parameter using Ant Colony Optimization (ACO) and was able to reduce the cost function of the system. In another study, Xu and Li (2019) anticipated the problem of calculating the total execution time of processes on a VM. They considered this problem as NP-hard and introduced a learning effect-based weighted model. Their model accurately estimates the total completion time and

TABLE 7 Comparison of parameters used in virtual machine scheduling.

References	Response Time	Makespan	Degree of Imbalance	Waiting Time	Execution Time	Energy	Performance	Latency	Execution cost	SLA	Bandwidth	Utilization	Fairness	Others
Rathore and Chana (2014)	√								√			√		
Salimi et al. (2012)			√				√							
Takouna et al. (2011)						√								√
Knauth and Fetzer (2012)						√								
Pegkas et al. (2018)	√									√				√
Imai et al. (2018)						√								
Bazarbayev et al. (2013)												√		
Lago et al. (2017)		√				√								
Hu et al. (2010)						√								
Xia et al. (2009)								√						√
Von Laszewski et al. (2009)			√			√								
Rao and Zhou (2014)				√						√			√	
Ding et al. (2020)						√			√					√
Quang-Hung and Thoai (2015)			√			√								
Xiao et al. (2014)						√						√		
Beloglazov and Buyya (2012)						√				√				
Kim et al. (2009)	√												√	
Zhao et al. (2014)						√				√				
Wu et al. (2018)					√		√							
Ebrahimirad et al. (2015)		√				√						√		
Saravanakumar and Arun (2016)												√		√
Kim et al. (2009)											√			
Miao and Chen (2015)							√							
Kertesz et al. (2016)			√				√							
Quesnel et al. (2013)					√				√					
Zhou and Yao (2017)												√		
Seo et al. (2014)						√								
Li X. et al. (2018)		√												√
Wu et al. (2018)						√				√				
Li et al. (2022)					√		√							

(Continued)

10.3389/fcomp.2024.1288552

TABLE 7 (Continued)

References	Response Time	Makespan	Degree of Imbalance	Waiting Time	Execution Time	Energy	Performance	Latency	Execution cost	SLA	Bandwidth	Utilization	Fairness	Others
Xu et al. (2012)												✓		
Xu et al. (2012)						✓								
Lago et al. (2017)			✓									✓		
Al-Dulaimy et al. (2018)						✓					✓			✓
Xu et al. (2012)						✓			✓			✓		✓
Yu et al. (2019)					✓				✓				✓	
Qiu et al. (2019)				✓			✓							
Kim et al. (2009)		✓				✓								
Xu et al. (2012)						✓	✓							
Xing et al. (2017)						✓							✓	
Hu et al. (2010)			✓						✓					
Kumar and Raza (2015)							✓					✓		
Cho et al. (2015)												✓		✓
Gondhi and Sharma (2015)						✓						✓		
Liu et al. (2017)					✓				✓					
Wang et al. (2018)		✓				✓								
Qin et al. (2019)								✓						
Xu and Li (2019)		✓			✓					✓				
Zhao et al. (2014)					✓									✓
Sui et al. (2019)						✓								
Li Y. et al. (2018)			✓									✓		
Feng and Zhu (2019)					✓									
Xu et al. (2012)						✓								
Wan et al. (2020)		✓			✓							✓		
Qi et al. (2020)		✓												
Saravanakumar et al. (2021)						✓						✓		
Xu et al. (2023)							✓		✓					
Karthikeyan and Soni (2020)						✓		✓						✓
Kruekaew and Kimpan (2020)		✓							✓					
Naik et al. (2020)						✓						✓		

(Continued)

10.3389/fcomp.2024.1288552

TABLE 7 (Continued)

References	Response Time	Makespan	Degree of Imbalance	Waiting Time	Execution Time	Energy	Performance	Latency	Execution cost	SLA	Bandwidth	Utilization	Fairness	Others
Rana et al. (2022)						✓								
Medara and Singh (2021)	✓		✓											✓
Ajmera and Tewari (2021)			✓			✓								
Chaudhury (2021)						✓						✓		
Alsadie (2021)						✓						✓		
Ss and Hs (2022)				✓										✓
Sheng (2022)		✓												✓

maximum lateness minimization. The proposed schedule-based rule exhibits better near-optimal results.

In another progress, Zhao et al. (2019) investigated an improved scheduling technique to reduce the high upfront cost of the systems. The proposed dynamic bin packing model used a divide and conquer strategy with a branch and bound algorithm (DCBB) for minimizing the VMs on the physical servers. The method is evaluated on three different real-time workloads and also on synthetic workloads. The experimental results show its superiority over comparative techniques for execution time and fast convergence rate.

By applying a machine learning technique for load balancing, Sui et al. (2019) established an intelligent technique for scheduling VMs in the data centers. First, the prediction is done for incoming workloads on the servers by utilizing a hybridization of the genetic algorithm with the combination of a Support Vector Machine (SVM) named SVR_GA. Then, to improve the local search capability, Differential Evaluation (DE)-based adaptive algorithm (ESA_DE) is utilized to overcome the problem of load balancing. When compared with the benchmark algorithms, the proposed method overtakes in terms of energy saving by minimizing the VM migration. An intelligent Genetic Algorithm (GA)-based metaheuristic technique is proposed for dynamic VM scheduling for optimum resource allocation. In this study, both memory and CPU utilization are considered equally for VM migration in the scheduling process. The study claims improvement in load balancing and resource utilization; however, the results are not mentioned in the Li et al. (2019). Similarly, Yao et al. (2019) implemented a GA-based Revivification-based prediction (ERP) model to estimate the execution time of applications on VMs. Then, another method ERPA is used to minimize the execution times for parallel and distributed applications running on the optimized set of VMs. The simulation results confirm better execution time for the selected VMs.

Karthikeyan and Soni (2020) proposed a hybrid GA, variable neighborhood search (VNS), and PSO to address the VM allocation problem, improving resource utilization and minimizing completion time. However, they did not mention how this algorithm improved the parameters. A similar study proposed an ABC-based scheduling algorithm, HABC, to reduce the average make span time of task allocation and the degree of load imbalance in the VMs. The algorithm is designed to work in both homogeneous and heterogeneous systems (Kruekaew and Kimpan, 2020). The fruit fly is combined with Cuckoo search to overcome the deficiency of local optima entrapment, perform better in local search, and find the optimal solution for VM mapping in the cloud data centers. The proposed method works well compared with similar techniques to reduce energy and resource leakage (Naik et al., 2020). Rana et al. (2022) combined WOA with DA to develop VM scheduling techniques in the cloud environment. This study uses WOA as a global optimizer to generate optimal solutions. In contrast, DA replaces the substandard solutions generated by WOA and improves the search speed in the local search space. Medara and Singh (2021) presented a solution for reducing energy consumption and resource utilization between workflow scheduling and VM scheduling in the data center. The method uses a nature-inspired water wave optimization (WWO) algorithm to find the optimal solution for VM migration on the host machines. An artificial

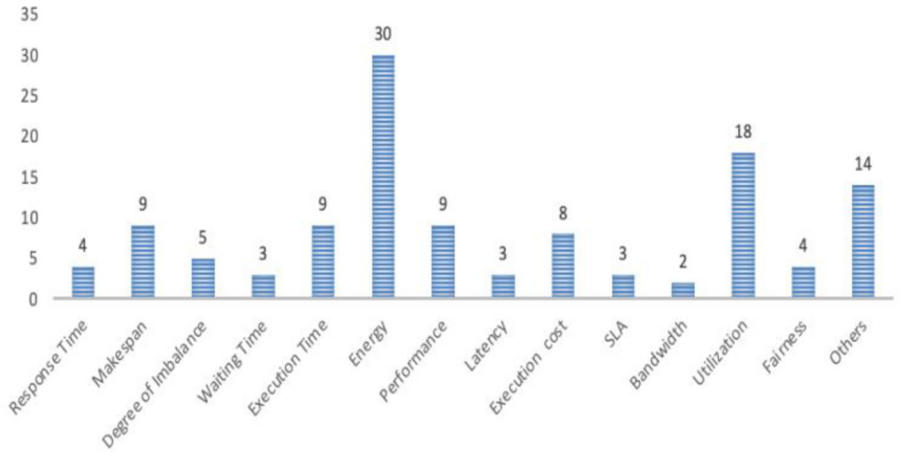

FIGURE 6
Number of parameters used in the reviewed literature.

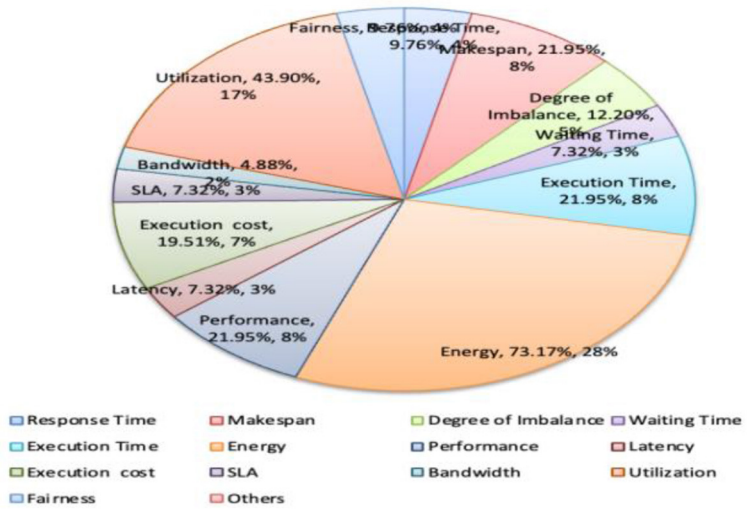

FIGURE 7
Percentage of virtual machine scheduling metrics in the reviewed literature.

immune-based clonal selection algorithm is modified to cope with the ever-changing cloud environment for VM scheduling. The randomized mutation operator is introduced to handle the dynamic load on the VM while scheduling. The simulation graphs show that the presented method performed better than benchmark methods for the energy reduction (Ajmera and Tewari, 2021).

In an identical work, Chaudhury (2021) proposed a metaheuristic-based scheduling algorithm for VM scheduling combining PSO and ACO. The proposed method retains the historical details of the scheduling components in its search process. It uses it to predict the incoming load on the cloud, reducing the load imbalance on the servers. Similarly, Alsadie (2021) modified the NSGA-II metaheuristic algorithm to cope with the dynamic environment of cloud scheduling. The technique works on two levels; first, the algorithm finds the optimal mapping solutions for tasks to the suitable VMs; second, the optimal solutions are generated for VM allocation to the best-fitted host in the data centers. The method outperforms other similar techniques but works only in a homogeneous environment. Because recent techniques do not consider NUMA architecture while designing VM scheduling, Sheng (2022) proposed multi-NUMA VM scheduling techniques by applying a machine learning approach. The authors first converted the VM scheduling problem into combinatorial optimization and then used reinforcement learning to guide the schedule per sample data. As per the result, the proposed techniques efficiently reduce the task allocation time on the host node.

Meta-heuristic approaches in VM scheduling have shown a remarkable ability to adapt and find near-optimal solutions in the dynamic landscape of cloud computing. Techniques such as the ACOPS algorithm, which combines Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), are particularly

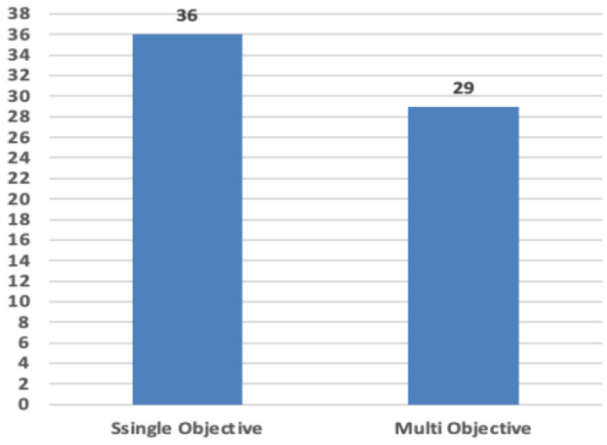

FIGURE 8
Comparison between single-objective and multi-objective studies in numbers.

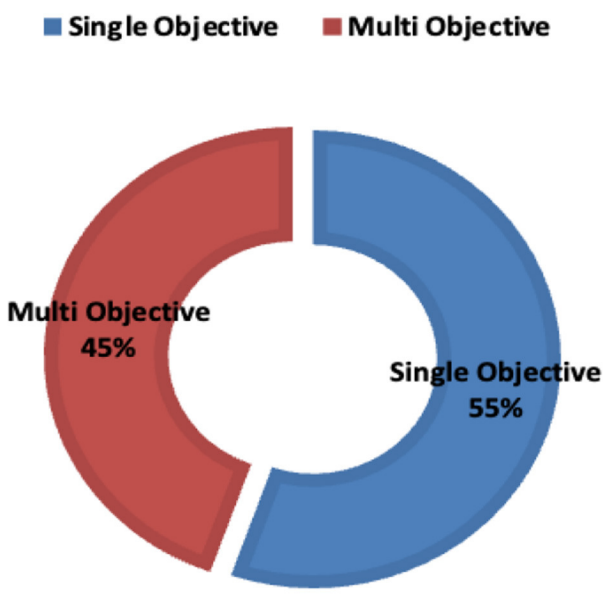

FIGURE 9
Percentage-based distribution between single-objective and multi-objective studies.

TABLE 8 Online available cloud datasets.

No.	Dataset/Workload	Url Source
1	OpenCloud Hadoop workload	http://ftp.pdl.cmu.edu
2	Eucalyptus IaaS cloud workload	https://www.cs.ucsb.edu/$\sim$rich/workload/
3	Yahoo cluster traces	https://webscope.sandbox.yahoo.com
4	TU Delft Bitbrains traces	http://gwa.ewi.tudelft.nl/datasets/
5	Cloud Dataset	https://archive.ics.uci.edu
6	Public Cloud Dataset	https://www.quora.com
7	Public Cloud Dataset	https://www.kdnuggets.com/
8	SEA dataset	http://www.schonlau.net/intrusion.html
9	Greenberg Dataset	http://saul.cpsc.ucalgary.ca
10	CERIT-SC grid workload	http://jsspp.org/workload/
11	RUU Dataset	http://sneakers.cs.columbia.edu
12	Public Cloud Dataset	http://www.cloudbus.org/workloads.html
13	Purdue University dataset	https://purr.purdue.edu/publications/datasets
14	CIDD Dataset	http://www.di.unipi.it/$\sim$hkholidy/projects/cidd/
15	Cloud computing services	https://data.europa.eu/euodp/data/dataset/
16	Open Nebula	https://opennebula.org/documentation/archives/
17	Python Library Dataset	https://www.python.org/downloads/
18	Dura Cloud	https://wiki.duraspace.org/
19	Azure	https://azure.microsoft.com/en-us/resources/
20	Rackspace	https://www.rackspace.com/en-gb
21	Google Cloud Traces	https://cloud.google.com/public-datasets/

noteworthy for their innovative use of historical server data to predict and adapt to changing workloads without the need for additional job information. Similarly, methods such as the Teaching and Learning Optimization (TLBO) algorithm have demonstrated significant energy savings by managing VMs in active and reserve modes to handle dynamic workloads efficiently.

The meta-heuristic methods stand out for their problem-solving versatility, with algorithms such as the Adaptive Penalty Function and Whale Optimization Algorithm (WOA) offering robust solutions under constraints such as time and cost. The parameters used in these studies, detailed in the surveyed literature, and reflected a focus on performance-power ratios, scalability, and energy conservation. Techniques, including hybrid models that integrate Genetic Algorithm (GA) with local search capabilities and nature-inspired algorithms such as the Water Wave Optimization (WWO), emphasize the importance of intelligent scheduling in enhancing resource utilization and reducing operational costs in cloud environments.

Furthermore, the application of meta-heuristic algorithms has transcended traditional boundaries, addressing the semi-sleep mode in VM scheduling to decrease resource request latency and optimizing execution time through learning-effect models. These advanced scheduling techniques, such as the hybrid GA and PSO or the incorporation of machine learning for load prediction, highlight the evolution of cloud resource management. They showcase a shift toward more intelligent, adaptive frameworks capable of meeting

the high demands of cloud services, reducing energy consumption, and ensuring cost-effective VM management while maintaining service level agreements and user satisfaction.

However, meta-heuristics are not without their limitations. Some of these methods may struggle with problem-specific parameter definition, leading to premature convergence and suboptimal solutions. There are challenges in maintaining the balance between exploration and exploitation, particularly in rapidly changing environments where static models fail to keep up.

In conclusion, while meta-heuristic methods have advanced the field of VM scheduling through their generalizability and ability to hybridize, they must continue to evolve to overcome their inherent weaknesses. Advancements in adaptive penalty functions and algorithmic hybridization show promise in enhancing time and cost constraints, but further innovation is needed to refine these methods for better accuracy, reducing energy consumption and efficient resource utilization in the dynamic and diverse realm of cloud computing.

6 VM scheduling in mobile edge computing

6.1 Mobile edge computing

The MEC, commonly known as multi-access computing or multi-access edge computing, is a distributed computing ecosystem that moves processing and data storage closer to the network's edge. It has been envisaged to prevent mobile devices from running heavy and power-hungry algorithms. Among other things, MEC is used to offload traffic off the leading network, allowing operators to save money while expanding network capacity (Pham et al., 2020). In the Internet of Things (IoT) context, MEC enables seamless integration of IoT and 5G (Qi et al., 2020).

6.2 Scheduling in MEC

In MEC, VM scheduling is essential for task offloading and resource allocations. Dynamic resource allocation uses Lyapunov optimization, a decision engine and deep-reinforcement learning. Priority scheduling is when tasks are scheduled based on their priority (Wei et al., 2017; Alfakih et al., 2020). The authors of the study mentioned in the reference (Mao et al., 2016; Gao and Moh, 2018) proposed joint offloading and priority-based task scheduling. The goal has been to reduce task completion time and the cost of edge server VM use. The same approach has been used in the study mentioned in the reference (Lei et al., 2019), where the authors extended further scope to include multi-users in a narrow-band IoT environment and solved the offloading using dynamic programming techniques. Cotask offloading and schedules have been investigated in the study mentioned in the reference (Chiang et al., 2020). The authors formulated the problem of cotask offloading as a non-linear program and solved it using the deep dual learning method. Similarly, Choi et al. (2019) present a deadline-aware task offloading algorithm for mobile edge computing environments. The algorithm is based on classifying tasks according to their latency requirements and offloading them to the most appropriate edge server. The algorithm is designed to minimize the overall completion time of the tasks while satisfying the deadlines and maximizing resource utilization.

Zhu et al. (2023) proposed a new approach for offloading in mobile edge computing that utilizes an improved multi-objective immune cloning algorithm. The goal of the proposed method is to enhance the efficiency of offloading by optimizing multiple objectives, including maximizing computational performance and minimizing energy consumption. This new approach aims to improve the parameters of computational performance and energy efficiency in mobile edge computing offloading. Similarly, Li et al. (2023) put forth a jointly non-cooperative game-based offloading and dynamic service migration approach in mobile edge computing. The approach uses game theory to optimize the performance of the system by making optimal offloading and migration decisions based on limited resources, such as bandwidth and computation capacity. Naouri et al. (2021) put forward a novel framework for mobile-edge computing that optimizes task offloading. The authors aim to address the challenges in offloading tasks from mobile devices to edge servers. The framework employs optimization techniques to improve the offloading decision-making process, leading to better performance and reduced energy consumption. The results show that the proposed framework outperforms existing solutions in terms of efficiency and effectiveness.

In the same vein, Cui et al. (2021) presented a new approach to task offloading scheduling for the application of mobile edge computing. The authors aim to improve the performance and efficiency of task offloading in mobile devices by proposing a new scheduling method. The approach considers various factors such as device resources, network conditions, and service requirements to make offloading decisions. The experimental results show that the proposed method outperforms existing solutions in terms of task completion time and energy consumption. Sheng et al. (2019) proposed a computation offloading strategy for mobile edge computing. The authors aim to optimize the offloading of computationally intensive tasks from mobile devices to edge servers. The proposed strategy takes into account various factors such as network conditions, device resources, and task requirements to make offloading decisions. The results show that the proposed strategy improves performance and reduces energy consumption compared with existing solutions. Hao et al. (2019) examined a formal concept analysis approach to VM scheduling in mobile edge computing. The authors aim to address the challenge of resource allocation in mobile devices when offloading tasks to edge servers. The proposed approach uses formal concept analysis to model the scheduling problem and find optimal solutions for task offloading.

Deadline-aware scheduling is another problem in which tasks are scheduled based on the time the task should be completed. Zhu et al. (2018) addressed the problem of scheduling multiple mobile devices under various MEC servers. Lakhan et al. (2022) devised an algorithm for scheduling fine-grained tasks in mobile edge computing environments. The algorithm considers both the tasks' deadlines and the edge servers' energy efficiency when scheduling the tasks. The algorithm aims to minimize the total energy consumption while satisfying the deadlines of the tasks

and maximizing resource utilization. The authors evaluate the proposed algorithm using simulations, and the results show that the algorithm outperforms existing algorithms in terms of energy efficiency and meeting deadlines. Ali and Iqbal (2022) proposed a task scheduling technique for offloading microservices-based applications in mobile cloud computing environments. The technique considers both the cost and energy efficiency when scheduling the tasks. The technique is designed to minimize the total cost while satisfying energy efficiency and meeting the deadlines of the tasks. The authors evaluate the proposed technique using simulations, and the results show that the technique outperforms existing techniques in terms of cost and energy efficiency. In the same vein, Bali et al. (2023) consider the priority of the tasks when scheduling tasks to offload data at edge and cloud servers. The technique is designed to minimize the total completion time while satisfying the priority and meeting the deadlines of the tasks. The authors evaluate the proposed technique using simulations, showing that the technique outperforms existing techniques in terms of completion time and meeting priority.

Qureshi et al. (2022) and Yadav and Sharma (2023) developed a method for improving the sustainability of mobile edge computing through blockchain technology. The presented method uses blockchain to secure cooperative task scheduling in these environments. The method aims to enhance task scheduling security by utilizing the blockchain's decentralized and immutable nature. The results show improved security and sustainability of task scheduling in mobile edge computing. The authors of Li et al. (2022) proposed a solution to enhance the efficiency of mobile edge computing by collaborating between User Plane Functions (UPFs) and edge servers. Their proposed algorithm, UPF selection, considers the current load and computing capacities of UPFs and edge servers for optimal resource utilization. The simulation results show that this approach improves system performance compared with traditional methods. In conclusion, the authors state that collaboration between UPFs and edge servers can significantly improve mobile edge computing performance. A different study is presented by Lou et al. (2023) on addressing the problem of scheduling dependent tasks in a mobile edge computing environment while considering the startup latency caused by limited bandwidth on edge servers. The authors propose a novel algorithm named Startup-aware Dependent Task Scheduling (SDTS), which selects the edge server with the earliest finish time for each dependent task. The selection process considers the edge servers' downloading workload, computation workload, and processing capability. Additionally, the algorithm employs a cloud clone for each task to utilize the scalable computation resources in the cloud. The results of simulations using real-world datasets show that SDTS outperforms existing baselines in terms of make span. In future study, the authors plan to further study the dependent task scheduling problem in more dynamic edge computing networks.

A scheduling and resource allocation technique for Mobile Edge Computing was proposed by Kuang et al. (2022) using the opposition-based Marine-Predator Algorithm. The method seeks to optimize the scheduling of multiple workflows and the allocation of resources in the mobile edge computing setting, balancing computation load and energy consumption. The opposition-based Marine-Predator Algorithm combines the marine-inspired and predator-prey algorithms, which are designed to effectively address the multi-objective optimization problem in mobile edge computing systems. Jian et al. (2022) presented a new high-efficiency learning model for VM placement in mobile edge computing. The model aims to optimize VM placement in a way that improves the system's efficiency, considering various factors, such as computational resources, network constraints, and other relevant variables. The authors describe how the proposed model utilizes machine learning techniques to dynamically adjust the placement of VMs based on real-time system conditions, resulting in a more efficient and effective mobile edge computing environment. Similarly, Hao et al. (2021) proposed a new energy-conscious scheduling method for edge computing using clustering techniques. The aim is to balance energy consumption and performance in edge devices. The method involves grouping edge devices based on their energy consumption characteristics and scheduling tasks accordingly. The results indicate that the proposed solution significantly improves energy efficiency while preserving performance compared with existing approaches. Alfakih et al. (2021) presented a multi-objective optimization technique for resource allocation in mobile edge computing using accelerated particle swarm optimization and dynamic programming. The authors aim to improve resource utilization in edge devices by considering multiple objectives, such as energy consumption, processing time, and cost. The proposed method balances these objectives to find optimal solutions for resource allocation. The results show that the proposed technique outperforms existing methods regarding efficiency and effectiveness.

6.3 Comparing VM scheduling and MEC

Virtual Machine scheduling in cloud computing and MEC are similar in that they both aim to allocate resources effectively and efficiently to multiple VMs running on a single physical host. However, there are some differences between the two which are found in the literature below.

6.3.1 Similarities

- Both focus on resource allocation: Both cloud and MEC aim to allocate physical resources, such as CPU, memory, and network bandwidth, to multiple VMs in a way that maximizes resource utilization and minimizes resource waste.
- Both use algorithms to schedule VMs: Both cloud and MEC use various scheduling algorithms to determine which VMs should run on which physical resources, based on factors such as priority, performance requirements, and resource availability.

6.3.2 Differences

- Scale: Cloud computing operates on a much larger scale compared with MEC, with data centers often serving thousands of users. In contrast, MEC operates at the edge of

the network, closer to end-users, with fewer VMs, and less overall computing power.
- Latency requirements: MEC is designed to provide low-latency services to users, whereas cloud computing is less concerned with latency. As a result, MEC often has more stringent requirements for VM scheduling and resource allocation, to meet its low-latency goals.
- Network connectivity: Cloud computing is typically located far from end-users, which is connected to them over a wide-area network (WAN). In contrast, MEC operates at the edge of the network, close to end-users, and is connected to them over a local area network (LAN). This difference affects the scheduling algorithms used and the types of resources that are available for allocation.

6.3.3 Common parameters in MEC

- Latency: The time taken for data to travel from the source to the destination. Low latency is critical in MEC to provide real-time services.
- Bandwidth: The amount of data can be transmitted per unit of time. High bandwidth is necessary to support data-intensive applications.
- Computing resources: The amount of processing power, memory, and storage available at the edge. This affects the ability of MEC to support complex applications and services.
- Energy consumption: The amount of power required to run MEC services. This is a critical factor in mobile devices with limited battery life.
- Availability: The degree to which MEC services are available to users. This can be affected by network conditions, system failures, and other factors.
- Security: The measures in place to protect MEC services from unauthorized access, hacking, and other security threats.
- Cost: The economic cost of deploying and operating MEC infrastructure and services.
- Scalability: The ability of an MEC system to handle increasing amount of data and devices over time.

6.4 Validity of the research

The SLR analyzes (see section 3.3) the existing literature on VM scheduling and presents a taxonomy of approaches to solving virtual scheduling problems. It tries to put forward the most significant solutions in the field of scheduling technique optimization to date. Although the authors have cautiously selected the most relevant articles and QAC processes from different reliable sources, there is still a potential threat to the validity of the work in the conduct, design, and analysis phases. To avoid the biasness in the exclusion and inclusion processes, the authors tried to search the maximum available literature. Even though, there is a possibility of oversight of some studies due to ambiguity in the literature, technical reports, and theses. This survey's stringent methodology serves as the study's proof of validity (see sections 3.4 and 3.5). The dissemination of the analysis of this study will allow the researchers to effectively utilize the results.

7 Future issues and opportunities

Despite the availability of a plethora of literature in the area of VM scheduling techniques, there remain several aspects that have not been addressed extensively and exhaustively. This is true in the case of problem formulation and the enhancement of techniques. Many authors have discussed the challenges and opportunities in this area with different aspects, whereas we emphasize the fundamental performance metrics and objectives of VM scheduling, allocation, and deallocation of resources. Moreover, we offer our thoughts on where the state-of-the-art algorithms and methods could go and how they could be improved upon. The following sections provide further explanation.

7.1 Recourse mapping problem

In the scheduling problem, the mapping of a task to VMs and VMs to PMs is treated as the formulation of the problem using several techniques. Notably, in the heterogeneous infrastructure, it becomes ubiquitous to examine the mapping of tasks to VMs. In general, the users are only interested to map their tasks efficiently and safely to PMs using VMs. However, the clearer distinction of the mapping at each level in the scheduling is crucial. Hence, the investigation for enhancement and development of tri-lateral scheduling techniques is an issue worth considering.

7.2 Energy-aware optimization

Although all the optimization techniques discussed in the study are essential, some of the techniques were found contradictory to each other. Some of the techniques consolidate the VMs and increase physical resources when workloads increase. The other techniques de-consolidate VMs in the case of overheating and put extra constraints on the nodes. Therefore, combining these two optimization techniques seems a daunting task to solve multi-objective problems. Existing techniques in VM scheduling use VM selection, VM placement, and VM migration methods. The selection of a method for designing a scheduling technique is crucial and needs a distinct understanding of the issue.

Moreover, some traditional techniques are implemented in server-level scheduling to address the same problem. For example, in Dynamic Voltage and Frequency Scaling (DVFS), individual-level components-based scheduling, the remaining nodes are switched off or put on sleep mode. On the network level, equipment such as routers and switches are also considered, making all these processes more complex. At both levels, the scheduling techniques mainly work on a static or fixed node in a controlled environment. Hence, more study is needed to explore and design efficient techniques, which can cater to both levels of the scheduling problem in a dynamic environment to support increased utilization and scalability of the recourses.

7.3 Complexity in server-level and network-level scheduling

Server-level and network-level scheduling in cloud environments pose intricate challenges due to the dynamic interplay of various components. At the server level, techniques such as Dynamic Voltage and Frequency Scaling (DVFS) allow dynamic adjustments of voltage and frequency based on computational load, yet they introduce complexities related to stability and thermal management. Furthermore, component-based scheduling, which manages individual server components like CPUs or memory, offers energy-saving opportunities but demands careful management of inter-component dependencies. On the network front, equipment-level management of routers, switches, and other devices necessitates balancing energy conservation and maintaining optimal performance, especially when considering the latency of reactivating equipment. A notable challenge is the contrast between static scheduling techniques and the inherent dynamism of real-world cloud environments, which is characterized by fluctuating traffic patterns and node variabilities. The optimal path forward lies in holistic solutions that merge server and network-level considerations, requiring algorithms capable of simultaneously managing VM placement, network routing, and component management. This intricate landscape underscores the need for comprehensive, adaptable scheduling solutions tailored to the multifaceted demands of modern cloud infrastructures.

7.4 VM scheduling in the context of IoT and industry 4.0

The burgeoning landscape of the Internet of Things (IoT) has led to an exponential surge in data generation, necessitating real-time or near-real-time processing. Efficient VM scheduling in cloud or edge environments becomes indispensable to handle this data deluge, ensuring timely and optimized data handling. Complementing this, the advent of edge computing emphasizes processing data closer to its source, mitigating the need for centralized cloud processing. In such edge contexts, dynamic VM scheduling becomes pivotal, ensuring real-time responses vital for applications ranging from autonomous vehicles to industrial sensor networks. This convergence of VM scheduling and edge processing finds its zenith in the realm of Industry 4.0, the Fourth Industrial Revolution. Embracing smart factories equipped with web-augmented machinery, Industry 4.0 underscores the seamless integration of the entire production chain, visualizing and autonomously making informed decisions. Herein, VMs play a crucial role, hosting analytics tools and platforms and processing data from these interconnected machines. Efficient VM scheduling ensures that these analytics tools consistently avail the necessary computational resources, facilitating the real-time analytics that are the cornerstone of Industry 4.0 paradigms.

7.5 Multi-objective optimization

Almost half of the literature focuses on solving a single-objective-optimization problem, as shown in Figure 9. Generally, the studies compare the research with some traditional, vague, and even obsolete techniques which seem to fall short, given the magnitude of the problems. Second, the majority of the mentioned studies focus on more common objective functions, such as makespan, energy, response time, waiting time, execution time, and load imbalance. The studies either completely ignore or lay inadequate stress on other important objectives, such as availability, throughput, recovery time, fairness, SLA, utilization, and fault tolerance. In addition, a major share of the literature studies is done on simulation-based tools using dummy datasets rather than real hypervisors, e.g., CloudSim, Xen, Open Nebula, and KVM. These studies tend to neglect the real traces in the real environment. Moreover, it is a much-needed stance of research to instigate future researchers to come out with efficient techniques which can focus on the real cloud environment for solving multi-objective problems.

7.6 Heuristics and mete-heuristics approach

VM scheduling is an NP-hard problem for which state-of-the-art algorithms are modified to find a good approximation to the ideal solution. That is to say, the resilience and acceptability of heuristic and meta-heuristic approaches to the scheduling problem are making their ground-breaking solutions to the problem. Many improved rule-based heuristics, e.g., First Come, First Serve Minimum Completion-Time, Minimum Execution-Time, Min-min, and Max-min have been proposed to resolve the problematic issues of cloud scheduling. These algorithms produce results faster than meta-heuristics algorithms in certain circumstances and achieve the optimal result through accuracy, completeness, and speed. Furthermore, several modified and hybrid nature-inspired algorithms are proposed based on modern algorithms, such as GA, ACO, and PSO, which have shown significant achievement in resolving single-objective and multi-objective problems. These algorithms perform better in multi-dimensional space than exact and approximation algorithms. However, there are more to be explored from the gems of the recently developed swarm-based mete-heuristics algorithms such as League Championship Algorithm (Kashan et al., 2021), Cuckoo Search (CS) (Saif et al., 2022), Krill Herd (KH) (Rahumath et al., 2021), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016; Rana et al., 2020), and Simulated Annealing (SA) (Tanha et al., 2021), to name a few.

7.7 Mobile edge computing

The future of MEC is expected to be characterized by increased integration with 5G networks, advanced edge AI capabilities, and more efficient and secure data processing. MEC will play a crucial role in the growth of the Internet of Things (IoT) and Industry

4.0 by enabling the processing of large amount of data generated by connected devices in real-time and providing the necessary control and feedback. MEC will also drive the development of virtual and augmented reality experiences, providing low-latency processing and high-speed connectivity. Additionally, MEC will facilitate the distribution of computing resources across the network edge, enabling a more flexible and scalable solution for various computing needs. With its ability to handle sensitive data and prevent cyber-attacks, MEC is expected to provide a more secure computing environment in the future. Overall, MEC is poised to play a significant role in shaping the future of computing and communication technology.

8 Conclusion

The study presented an SLR of VM scheduling techniques in cloud and mobile computing. The study follows a rigorous protocol to select the most relevant works from the literature for this study. The SLR analyzed 67 articles out of 722 and presented the outcome for future researchers. The study answered three research questions as per collected data and the experience earned throughout the research. The first research question highlights the importance of VM scheduling and its possible contribution to the growth of cloud systems. The second question evaluates the performance of existing scheduling approaches in meeting the target of VM scheduling matrices. Finally, the third research question attempts to comprehend the role of VM scheduling in solving recent optimization problems and disseminates the challenges and future directions. Moreover, the SLR includes the most relevant articles addressing MEC scheduling and analyzes the contemporary trends, similarities, and differences with VM scheduling in a cloud environment.

In addition, the study highlights the current scheduling techniques' strengths and weaknesses and classifies the possible solutions into three conventional methods: heuristics methods and meta-heuristic methods. It also critically analyzes the most common performance metrics used in VM scheduling in MEC and cloud computing. This study asserted that VM scheduling techniques in Cloud and MEC are indispensable as they let us introduce new paradigms in cloud scheduling. These developments significantly increase resource utilization, processing power, latency, and network connectivity. The authors anticipate that this survey will help practitioners and academics select the most appropriate literature and utilize it as a reference point in their research to solve cloud scheduling problems.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

NR: Conceptualization, Methodology, Writing – original draft. FJ: Formal analysis, Writing – review & editing. ZK: Formal analysis, Visualization, Writing – review & editing. WA: Supervision, Writing – review & editing, Funding acquisition. IB: Supervision, Resources, Visualization, Writing – review & editing. MH: Data curation, Software, Validation, Writing – review & editing. MU: Supervision, Investigation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This open access research was supported by Qatar National Library (QNL).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adhikary, T., Das, A. K., Razzaque, M. A., and Sarkar, A. J. (2013). "Energy-efficient scheduling algorithms for data center resources in cloud computing," in *2013 IEEE 10th International Conference on High Performance Computing and Communications and 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. IEEE, 1715–1720.

Ahmad, R. W., Gani, A., Ab. Hamid, S. H., Shiraz, M., Xia, F., and Madani, S. A. (2015a). Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. *J. Supercomput.* 71, 2473–2515. doi: 10.1007/s11227-015-1400-5

Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., Xia, F., et al. (2015b). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *J. Netw. Comput. Appl.* 52, 11–25. doi: 10.1016/j.jnca.2015.02.002

Ahmad, S., Hang, L., and Kim, D. H. (2017). Design and implementation of application programming interface for Internet of things cloud. *Int. J. Netw. Manage.* 27:1936. doi: 10.1002/nem.1936

Aikat, J., Akella, A., Chase, J. S., Juels, A., Reiter, M. K., Ristenpart, T., et al. (2017). Rethinking security in the era of cloud computing. *IEEE Secur. Privacy* 15, 60–69. doi: 10.1109/MSP.2017.80

Ajmera, K., and Kumar Tewari, T. (2024). Dynamic virtual machine scheduling using residual optimum power-efficiency in the cloud data center. *The Comput. J.* 67, 1099–1110. doi: 10.1093/comjnl/bxad045

Ajmera, K., and Tewari, T. K. (2021). VMS-MCSA: virtual machine scheduling using modified clonal selection algorithm. *Cluster Comput.* 24, 3531–3549. doi: 10.1007/s10586-021-03320-5

Ajmera, K., and Tewari, T. K. (2023). SR-PSO: server residual efficiency-aware particle swarm optimization for dynamic virtual machine scheduling. *J. Supercomput.* 79, 15459–15495. doi: 10.1007/s11227-023-05270-8

Al-Dulaimy, A., Itani, W., Zantout, R., and Zekri, A. (2018). Type-aware virtual machine management for energy efficient cloud data centers. *Sust. Comput. Inf. Syst.* 19, 185–203. doi: 10.1016/j.suscom.2018.05.012

Alfakih, T., Hassan, M. M., and Al-Razgan, M. (2021). Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing. *IEEE Access* 9, 167503–167520. doi: 10.1109/ACCESS.2021.3134941

Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C., and Fortino, G. (2020). Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA. *IEEE Access* 8, 54074–54084. doi: 10.1109/ACCESS.2020.2981434

Ali, A., and Iqbal, M. M. (2022). A cost and energy efficient task scheduling technique to offload microservices based applications in mobile cloud computing. *IEEE Access* 10, 46633–46651. doi: 10.1109/ACCESS.2022.3170918

Alsadie, D. (2021). A metaheuristic framework for dynamic virtual machine allocation with optimized task scheduling in cloud data centers. *IEEE Access* 9, 74218–74233. doi: 10.1109/ACCESS.2021.3077901

Bali, M. S., Gupta, K., Gupta, D., Srivastava, G., Juneja, S., Nauman, A., et al. (2023). An effective technique to schedule priority aware tasks to offload data on edge and cloud servers. *Measur. Sensors* 26:100670. doi: 10.1016/j.measen.2023.100670

Bazarbayev, S., Hiltunen, M., Joshi, K., Sanders, W. H., and Schlichting, R. (2013). "Content-based scheduling of virtual machines (VMs) in the cloud," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 93–101.

Beloglazov, A., Abawajy, J., and Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Fut. Gener. Comput. Syst.* 28, 755–768. doi: 10.1016/j.future.2011.04.017

Beloglazov, A., and Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr. Comput. Prac. Exp.* 24, 1397–1420. doi: 10.1002/cpe.1867

Bouterse, B., and Perros, H. (2017). Dynamic VM allocation in a SaaS environment. *Annal. Telecommunications* 2017, 1–14. doi: 10.1007/s12243-017-0589-0

Buyya, R., and Ranjan, R. (2010). Special section: federated resource management in grid and cloud computing systems. *Fut. Gen. Comput. Syst.* 26, 1189–1191. doi: 10.1016/j.future.2010.06.003

Challita, S., Paraiso, F., and Merle, P. (2017). "Towards formal-based semantic interoperability in multi-clouds: the fclouds framework," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE, 710–713.

Charband, Y., and Navimipour, N. J. (2016). Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research. *Inf. Syst. Front.* 18, 1131–1151. doi: 10.1007/s10796-016-9628-z

Chaudhury, K. S. (2021). A particle swarm and ant Colony optimization based load balancing and virtual machine scheduling algorithm for cloud computing environment. *TURCOMAT* 12, 3885–3898. doi: 10.17762/turcomat.v12i11.6504

Chauhan, N., Rakesh, N., and Matam, R. (2018). "Assessment on VM placement and VM selection strategies," in *Nature Inspired Computing*. Springer, 157–163.

Chiang, Y. H., Chiang, T. W., Zhang, T., and Ji, Y. (2020). Deep-dual-learning-based cotask processing in multiaccess edge computing systems. *IEEE Int. Things J.* 7, 9383–9398. doi: 10.1109/JIOT.2020.3004165

Cho, K. M., Tsai, P. W., Tsai, C. W., and Yang, C. S. (2015). A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. *Neural Comput. Appl.* 26, 1297–1309. doi: 10.1007/s00521-014-1804-9

Choi, H., Yu, H., and Lee, E. (2019). Latency-classification-based deadline-aware task offloading algorithm in mobile edge computing environments. *Applied Sci.* 9:4696. doi: 10.3390/app9214696

Corradi, A., Fanelli, M., and Foschini, L. (2014). VM consolidation: a real case based on OpenStack Cloud. *Fut. Gener. Comp. Syst.* 32, 118–127. doi: 10.1016/j.future.2012.05.012

Cui, Y., Zhang, D., Zhang, T., Yang, P., and Zhu, H. (2021). "A new approach on task offloading scheduling for application of mobile edge computing," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q., and Zeng, J. (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Gener. Comput. Syst.* 108, 361–371. doi: 10.1016/j.future.2020.02.018

Duan, J., and Yang, Y. (2017). A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Trans. Parallel Distrib. Syst.* 28, 2131–2144.

Ebrahimirad, V., Goudarzi, M., and Rajabi, A. (2015). Energy-aware scheduling for precedence-constrained parallel virtual machines in virtualized data centers. *J. Grid Comput.* 13, 233–253. doi: 10.1007/s10723-015-9327-x

Feng, Y., and Zhu, Y. (2019). "Pes: proactive event scheduling for responsive and energy-efficient mobile web computing," in *Proceedings of the 46th International Symposium on Computer Architecture*, 66–78.

Gao, L., and Moh, M. (2018). "Joint computation offloading and prioritized scheduling in mobile edge computing," in *2018 International Conference on High Performance Computing and Simulation (HPCS)*. IEEE.

Gondhi, N. K., and Sharma, A. (2015). "Local search based ant colony optimization for scheduling in cloud computing," in *Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on 2015*. IEEE.

Hao, F., Pang, G., Pei, Z., Qin, K., Zhang, Y., Wang, X., et al. (2019). Virtual machines scheduling in mobile edge computing: a formal concept analysis approach. *IEEE Trans. Sust. Comput.* 5, 319–328. doi: 10.1109/TSUSC.2019.2894136

Hao, Y., Cao, J., Wang, Q., and Du, J. (2021). Energy-aware scheduling in edge computing with a clustering method. *Future Gen. Comput. Syst.* 117, 259–272. doi: 10.1016/j.future.2020.11.029

Hu, J., Gu, J., Sun, G., and Zhao, T. (2010). "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *2010 3rd International Symposium on Parallel Architectures, Algorithms and Programming* (IEEE), 89–96.

Hu, L., Jin, H., Liao, X., Xiong, X., and Liu, H. (2008). "Magnet: A novel scheduling policy for power reduction in cluster with virtual machines," in *2008 IEEE International Conference on Cluster Computing*. IEEE, 13–22.

Imai, S., Patterson, S., and Varela, C. A. (2018). "Uncertainty-aware elastic virtual machine scheduling for stream processing systems," in *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE.

Jian, C., Bao, L., and Zhang, M. (2022). A high-efficiency learning model for virtual machine placement in mobile edge computing. *Cluster Comput.* 25, 3051–3066. doi: 10.1007/s10586-022-03550-1

Kalra, M., and Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egypt. Inf. J.* 16, 275–295. doi: 10.1016/j.eij.2015.07.001

Karthikeyan, P., and Soni, R. (2020). A hybrid PSO optimised virtual machine scheduling algorithm in cloud computing. *Int. J. Bus. Inf. Syst.* 34, 536–559. doi: 10.1504/IJBIS.2020.109028

Kashan, A. H., Balavand, A., Karimiyan, S., and Soleimani, F. (2021). "The league championship algorithm: applications and extensions," in *Handbook of AI-based Metaheuristics* (London: CRC Press), 201–218.

Kertesz, A., Dombi, J., and Benyi, A. (2016). A pliant-based virtual machine scheduling solution to improve the energy efficiency of iaas clouds. *J. Grid Comput.* 14, 41–53. doi: 10.1007/s10723-015-9336-9

Khan, M. A., Paplinski, A., Khan, A. M., Murshed, M., and Buyya, R. (2018). Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: a review. *Sust. Cloud Energ. Serv. Princip. Prac.* 22, 135–165. doi: 10.1007/978-3-319-62238-5_6

Khosravi, A., Nadjaran Toosi, A., and Buyya, R. (2017). Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers. *Concurr. Comput. Prac. Exp.* 29:e4125. doi: 10.1002/cpe.4125

Kim, H., Lim, H., Jeong, J., Jo, H., and Lee, J. (2009). "Task-aware virtual machine scheduling for I/O performance," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, 101–110.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele Univ.* 33, 1–26.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., Linkman, S., et al. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Inf. Softw. Technol.* 51, 7–15. doi: 10.1016/j.infsof.2008.09.009

Kitchenham, B., Charters, S., Budgen, D., Brereton, P., Turner, M., Linkman, S., et al. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Available online at: https://userpages.uni-koblenz.de/~laemmel/esecourse/slides/slr.pdf

Knauth, T., and Fetzer, C. (2012). "Energy-aware scheduling for infrastructure clouds in Cloud Computing Technology and Science (CloudCom)," in *2012 IEEE 4th International Conference on 2012*. IEEE.

Kruekaew, B., and Kimpan, W. (2020). Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing. *Int. J. Comput. Int. Syst.* 13, 496–510. doi: 10.2991/ijcis.d.200410.002

Kuang, F., Xu, Z., and Masdari, M. (2022). Multi-workflow scheduling and resource provisioning in mobile edge computing using opposition-based marine-predator algorithm. *Perv. Mobile Comput.* 87:101715. doi: 10.1016/j.pmcj.2022.101715

Kumar, D., and Raza, Z. (2015). "A PSO based VM resource scheduling model for cloud computing," in Computational Intelligence and Communication Technology (CICT), 2015 IEEE International Conference on 2015. IEEE.

Kumar, M., Sharma, S. C., Goel, A., and Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. *J. Netw. Comput. Appl.* 143, 1–33. doi: 10.1016/j.jnca.2019.06.006

Lago, D. G., Madeira, E. R., and Medhi, D. (2017). Energy-aware virtual machine scheduling on data centers with heterogeneous bandwidths. *IEEE Trans. Parallel Distrib. Syst.* 29, 83–98. doi: 10.1109/TPDS.2017.2753247

Lakhan, A., Mohammed, M. A., Rashid, A. N., Kadry, S., and Abdulkareem, K. H. (2022). Deadline aware and energy-efficient scheduling algorithm for fine-grained tasks in mobile edge computing. *Int. J. Web Grid Serv.* 18, 168–193. doi: 10.1504/IJWGS.2022.121935

Leelipushpam, P. G. J., and Sharmila, J. (2013). "Live VM migration techniques in cloud environment—a survey. in Information and Communication Technologies (ICT)," in *2013 IEEE Conference on 2013*. IEEE.

Lei, L., Xu, H., Xiong, X., Zheng, K., and Xiang, W. (2019). Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system. *IEEE Int. Things J.* 6, 5345–5362. doi: 10.1109/JIOT.2019.2900550

Li, C., Zhang, Q., and Luo, Y. (2023). A jointly non-cooperative game-based offloading and dynamic service migration approach in mobile edge computing. *Know. Inf. Syst.* 65, 2187–2223. doi: 10.1007/s10115-022-01822-1

Li, H., Zhu, G., Zhao, Y., Dai, Y., and Tian, W. (2017). Energy-efficient and QoS-aware model based resource consolidation in cloud data centers. *Cluster Comput.* 20, 2793–2803. doi: 10.1007/s10586-017-0893-5

Li, J., Yang, S., Wang, J., and Yang, L. (2019). Research on dynamic virtual machine scheduling strategy based on improved genetic algorithm. *J. Phys. Conf. Series* 1168:052014. doi: 10.1088/1742-6596/1168/5/052014

Li, W., Liu, X., Zhang, X., and Zhang, X. (2017). "Multi-resource fair allocation with bounded number of tasks in cloud computing systems," in *Theoretical Computer Science: 35th National Conference, NCTCS 2017, Wuhan, China, October 14-15, 2017, Proceedings*. Springer Singapore, 3–17.

Li, X., Garraghan, P., Jiang, X., Wu, Z., and Xu, J. (2017). Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. *IEEE Trans. Parallel Distrib. Syst.* 29, 1317–1331. doi: 10.1109/TPDS.2017.2688445

Li, X., Jiang, X., Garraghan, P., and Wu, Z. (2018). Holistic energy and failure aware workload scheduling in Cloud datacenters. *Fut. Gener. Comput. Syst.* 78, 887–900. doi: 10.1016/j.future.2017.07.044

Li, Y., Li, W., and Jiang, C. (2010). "A survey of virtual machine system: current technology and future trends," in *2010 Third International Symposium on Electronic Commerce and Security*. *IEEE*, 332–336.

Li, Y., Wang, S., Hong, X., and Li, Y. (2018). "Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm," in *2018 37th Chinese Control Conference (CCC)* (IEEE), 4489–4494.

Li, Y., Zhou, A., Ma, X., and Wang, S. (2022). "Collaborative mobile edge computing through UPF selection," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Cham: Springer Nature Switzerland, 345–362.

Li, Z., Zhang, H., O'Brien, L., Cai, R., and Flint, S. (2013). On evaluating commercial cloud services: a systematic review. *J. Syst. Softw.* 86, 2371–2393. doi: 10.1016/j.jss.2013.04.021

Liu, L., Zhang, M., Buyya, R., and Fan, Q. (2017). Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing. *Concurr. Comput. Prac. Exp.* 29:e3942. doi: 10.1002/cpe.3942

Lou, J., Tang, Z., Jia, W., Zhao, W., and Li, J. (2023). Startup-aware dependent task scheduling with bandwidth constraints in edge computing. *IEEE Trans. Mobile Comput.* 12:868. doi: 10.1109/TMC.2023.3238868

Madni, S. H. H., Abd Latiff, M. S., and Coulibaly, Y. (2016). An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. *Indian J. Sci. Technol.* 9:80561. doi: 10.17485/ijst/2016/v9i4/80561

Madni, S. H. H., Latiff, M. S. A., and Coulibaly, Y. (2017). Recent advancements in resource allocation techniques for cloud computing environment: a systematic review. *Cluster Comput.* 20, 2489–2533. doi: 10.1007/s10586-016-0684-4

Manvi, S. S., and Shyam, G. K. (2014). Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. *J. Network Comput. Appl.* 41, 424–440. doi: 10.1016/j.jnca.2013.10.004

Mao, Y., Zhang, J., and Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Selected Areas Commun.* 34, 3590–3605. doi: 10.1109/JSAC.2016.2611964

Medara, R., and Singh, R. S. (2021). Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization. *Sim. Modelling Prac. Theor.* 110:102323. doi: 10.1016/j.simpat.2021.102323

Miao, T., and Chen, H. (2015). FlexCore: dynamic virtual machine scheduling using VCPU ballooning. *Tsinghua Sci. Technol.* 20, 7–16. doi: 10.1109/TST.2015.7040515

Milani, A. S., and Navimipour, N. J. (2016). Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends. *J. Netw. Comput. Appl.* 71, 86–98. doi: 10.1016/j.jnca.2016.06.003

Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Software* 95, 51–67. doi: 10.1016/j.advengsoft.2016.01.008

Mousavi, S., Mosavi, A., and Varkonyi-Koczy, A. R. (2018). "A load balancing algorithm for resource allocation in cloud computing," in *Recent Advances in Technology Research and Education: Proceedings of the 16th International Conference on Global Research and Education Inter-Academia 2017 16*. Springer International Publishing, 289–296.

Mukherjee, I., and Ray, P. K. (2006). A review of optimization techniques in metal cutting processes. *Comput. Ind. Eng.* 50, 15–34. doi: 10.1016/j.cie.2005.10.001

Mustafa, S., Nazir, B., Hayat, A., and Madani, S. A. (2015). Resource management in cloud computing: taxonomy, prospects, and challenges. *Comput. Electr. Eng.* 47, 186–203. doi: 10.1016/j.compeleceng.2015.07.021

Naik, B. B., Singh, D., and Samaddar, A. B. (2020). FHCS: Hybridised optimisation for virtual machine migration and task scheduling in cloud data center. *IET Commun.* 14, 1942–1948. doi: 10.1049/iet-com.2019.1149

Naouri, A., Wu, H., Nouri, N. A., Dhelim, S., and Ning, H. (2021). A novel framework for mobile-edge computing by optimizing task offloading. *IEEE Int. Things J.* 8, 13065–13076. doi: 10.1109/JIOT.2021.3064225

Navimipour, N. J., and Charband, Y. (2016). Knowledge sharing mechanisms and techniques in project teams: literature review, classification, and current trends. *Comput. Human Behav.* 62, 730–742. doi: 10.1016/j.chb.2016.05.003

Patel, K. S., and Sarje, A. K. (2012). "VM provisioning method to improve the profit and SLA violation of cloud service providers in Cloud Computing in Emerging Markets (CCEM)," in *2012 IEEE International Conference on 2012*. IEEE.

Pegkas, A., Alexakos, C., and Likothanassis, S. (2018). "Credit-based algorithm for virtual machines scheduling," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE.

Pham, Q. V., Fang, F., Ha, V. N., Piran, M. J., Le, M., Le, L. B., et al. (2020). A survey of multi-access edge computing in 5G and beyond: fundamentals, technology integration, and state-of-the-art. *IEEE Access* 8, 116974–117017. doi: 10.1109/ACCESS.2020.3001277

Prajapati, K. D. (2013). Comparison of virtual machine scheduling algorithms in cloud computing. *Int. J. Comput. Appl.* 83:2914. doi: 10.5120/14523-2914

Qi, L., Chen, Y., Yuan, Y., Fu, S., Zhang, X., Xu, X., et al. (2020). A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. *World Wide Web* 23, 1275–1297. doi: 10.1007/s11280-019-00684-y

Qin, B., Jin, S., and Zhao, D. (2019). Energy-efficient virtual machine scheduling strategy with semi-sleep mode on the cloud platform. *Int. J. Innov. Comput. Inf. Control* 15, 337–349. doi: 10.24507/ijicic.15.01.337

Qiu, Y., Jiang, C., Wang, Y., Ou, D., Li, Y., Wan, J., et al. (2019). Energy aware virtual machine scheduling in data centers. *Energies* 12:646. doi: 10.3390/en12040646

Quang-Hung, N., and Thoai, N. (2015). "Energy-efficient VM scheduling in IaaS Clouds," in *International Conference on Future Data and Security Engineering*. Springer.

Quesnel, F., Lèbre, A., Pastor, J., Südholt, M., and Balouek, D. (2013). "Advanced validation of the dvms approach to fully distributed vm scheduling," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 1249–1256.

Qureshi, M. B., Qureshi, M. S., Tahir, S., Anwar, A., Hussain, S., Uddin, M., et al. (2022). Encryption techniques for smart systems data security offloaded to the cloud. *Symmetry* 14:695. doi: 10.3390/sym14040695

Rahbari, D. (2022). Analyzing meta-heuristic algorithms for task scheduling in a fog-based IoT application. *Algorithms* 15:397. doi: 10.3390/a15110397

Rahimikhanghah, A., Tajkey, M., Rezazadeh, B., and Rahmani, A. M. (2022). Resource scheduling methods in cloud and fog computing environments: a systematic literature review. *Cluster Comput.* 15, 1–35. doi: 10.1007/s10586-021-03467-1

Rahumath, A. S., Natarajan, M., and Malangai, A. R. (2021). Resource scalability and security using entropy based adaptive krill herd optimization for auto scaling in cloud. *Wireless Pers. Commun.* 119, 791–813. doi: 10.1007/s11277-021-08238-0

Rana, N., and Abd Latiff, M. S. (2018). A cloud-based conceptual framework for multi-objective virtual machine scheduling using whale optimization algorithm. *Int. J. Innov. Comput.* 8:199. doi: 10.11113/ijic.v8n3.199

Rana, N., Abd Latiff, M. S., Abdulhamid, S. I. M., and Misra, S. (2022). A hybrid whale optimization algorithm with differential evolution optimization for multi-objective virtual machine scheduling in cloud computing. *Eng. Optim.* 54, 1999–2016. doi: 10.1080/0305215X.2021.1969560

Rana, N., Latiff, M. S. A., Abdulhamid, S. I. M., and Chiroma, H. (2020). Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments. *Neural Comput. Appl.* 32, 16245–16277. doi: 10.1007/s00521-020-04849-z

Rao, J., and Zhou, X. (2014). Towards fair and efficient SMP virtual machine scheduling. *ACM SIGPLAN Notices* 49, 273–286. doi: 10.1145/2692916.2555246

Rathore, N., and Chana, I. (2014). Load balancing and job migration techniques in grid: a survey of recent trends. *Wireless Pers. Commun.* 79, 2089–2125. doi: 10.1007/s11277-014-1975-9

Rodriguez, M. A., and Buyya, R. (2017). A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments. *Concurr. Comput. Prac. Exp.* 29:4041. doi: 10.1002/cpe.4041

Saif, M. A. N., Niranjan, S. K., and Murshed, B. A. H. (2022). "Multi-objective cuckoo search optimization algorithm for optimal resource allocation in cloud environment," in *2022 3rd International Conference for Emerging Technology (INCET)*. IEEE, 1–7.

Salimi, H., Najafzadeh, M., and Sharifi, M. (2012). Advantages, challenges and optimizations of virtual machine scheduling in cloud computing environments. *Int. J. Comput. Theor. Eng.* 4:189. doi: 10.7763/IJCTE.2012.V4.448

Saravanakumar, C., and Arun, C. (2016). Efficient idle virtual machine management for heterogeneous cloud using common deployment model. *KSII Trans. Int. Inf. Syst.* 10:2. doi: 10.3837/tiis.2016.04.002

Saravanakumar, C., Geetha, M., Manoj Kumar, S., Manikandan, S., Arun, C., Srivatsan, K., et al. (2021). An efficient technique for virtual machine clustering and communications using task-based scheduling in cloud computing. *Sci. Progr.* 2021, 1–15. doi: 10.1155/2021/5586521

Sayadnavard, M. H., Haghighat, A. T., and Rahmani, A. M. (2022). A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers. *Eng. Sci. Technol. Int. J.* 26:100995. doi: 10.1016/j.jestch.2021.04.014

Seo, J., Tchamgoue, G. M., and Kim, K. H. (2014). "Power-aware real-time virtual machine schedulers in discrete DVFS systems," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing* (IEEE), 459–463.

Sharifi, M., Salimi, H., and Najafzadeh, M. (2012). Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. *J. SUPERCOMPUT.* 61, 46–66. doi: 10.1007/s11227-011-0658-5

Shaw, S. B., and Singh, A. K. (2014). "A survey on scheduling and load balancing techniques in cloud computing environment," in *2014 International Conference on Computer and Communication Technology (ICCCT)* (IEEE), 87–95.

Sheng, J. (2022). Learning to schedule multi-NUMA virtual machines via reinforcement learning. *Pattern Recog.* 121, 108254. doi: 10.1016/j.patcog.2021.108254

Sheng, J., Hu, J., Teng, X., Wang, B., and Pan, X. (2019). Computation offloading strategy in mobile edge computing. *Information* 10:191. doi: 10.3390/info10060191

Ss, V. C., and Hs, A. (2022). Nature inspired meta heuristic algorithms for optimization problems. *Computing* 104, 251–269. doi: 10.1007/s00607-021-00955-5

Sui, X., Liu, D., Li, L., Wang, H., and Yang, H. (2019). Virtual machine scheduling strategy based on machine learning algorithms for load balancing. *EURASIP J. Wireless Commun. Netw.* 2019:160. doi: 10.1186/s13638-019-1454-9

Takouna, I., Dawoud, W., and Meinel, C. (2011). "Efficient virtual machine scheduling-policy for virtualized heterogeneous multicore systems," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2011), Las Vegas, NV*.

Tanha, M., Hosseini Shirvani, M., and Rahmani, A. M. (2021). A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments. *Neural Comput. Appl.* 33, 16951–16984. doi: 10.1007/s00521-021-06289-9

Uddin, M., Alsaqour, R., Shah, A., and Saba, T. (2014). Power usage effectiveness metrics to measure efficiency and performance of data centers. *Appl. Mathematics Inf. Sci.* 8:2207. doi: 10.12785/amis/080514

Uddin, M., Khalique, A., Jumani, A. K., Ullah, S. S., and Hussain, S. (2021). Next-generation blockchain-enabled virtualized cloud security solutions: review and open challenges. *Electronics* 10:2493. doi: 10.3390/electronics10202493

Uddin, M., Memon, J., Alsaqour, R., Shah, A., and Rozan, M. Z. A. (2015). Mobile agent based multi-layer security framework for cloud data centers. *Indian J. Sci. Technol.* 8:1. doi: 10.17485/ijst/2015/v8i12/52923

Von Laszewski, G., Wang, L., Younge, A. J., and He, X. (2009). "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, 1–10.

Wan, B., Dang, J., Li, Z., Gong, H., Zhang, F., Oh, S., et al. (2020). Modeling analysis and cost-performance ratio optimization of virtual machine scheduling in cloud computing. *IEEE Trans. Parallel Distrib. Syst.* 31, 1518–1532. doi: 10.1109/TPDS.2020.2968913

Wang, B., Jin, S., and Qin, B. (2018). Batch arrival based performance evaluation of a VM scheduling strategy in cloud computing. *Int. J. Innov. Comput. Inf. Control* 14, 455–467.

Wei, X., Wang, S., Zhou, A., Xu, J., Su, S., Kumar, S., et al. (2017). "MVR: An architecture for computation offloading in mobile edge computing," in 2017 *IEEE International Conference on Edge Computing (EDGE)*. IEEE, 232–235.

Wu, J., Lu, S., and Zheng, H. (2018). "On maximum elastic scheduling of virtual machines for cloud-based data center networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

Xia, Y., Niu, Y., Zheng, Y., Jia, N., Yang, C., Cheng, X., et al. (2008). "Analysis and enhancement for interactive-oriented virtual machine scheduling," in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. IEEE, 393–398.

Xia, Y., Yang, C., and Cheng, X. (2009). "PaS: a preemption-aware scheduling interface for improving interactive performance in consolidated virtual machine environment," in *2009 15th International Conference on Parallel and Distributed Systems* (IEEE), 340–347.

Xiao, P., Hu, Z., Liu, D., Zhang, X., and Qu, X. (2014). Energy-efficiency enhanced virtual machine scheduling policy for mixed workloads in cloud environments. *Comput. Electr. Eng.* 40, 1650–1665. doi: 10.1016/j.compeleceng.2014.03.002

Xie, X., Cao, W., Jin, H., Ke, X., and Luo, S. (2014). Design and implementation of process-aware predictive scheduling scheme for virtual machine. *The J. Supercomputing* 70, 1577–1587. doi: 10.1007/s11227-014-1254-2

Xing, G., Xu, X., Xiang, H., Xue, S., Ji, S., Yang, J., et al. (2017). Fair energy-efficient virtual machine scheduling for Internet of Things applications in cloud environment. *Int. J. Distrib. Sensor Networks* 13:1550147717694890. doi: 10.1177/1550147717694890

Xu, C., Gamage, S., Rao, P. N., Kangarlou, A., Kompella, R. R., and Xu, D. (2012). "vSlicer: latency-aware virtual machine scheduling via differentiated-frequency CPU slicing," in *Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing*, 3–14.

Xu, F., Liu, F., Jin, H., and Vasilakos, A. V. (2013). Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions. *Proc. IEEE* 102, 11–31. doi: 10.1109/JPROC.2013.2287711

Xu, H., and Li, X. (2019). Methods for virtual machine scheduling with uncertain execution times in cloud computing. *Int. J. Machine Learn. Cybernetics* 10, 325–335. doi: 10.1007/s13042-017-0717-1

Xu, H., Liu, Y., Wei, W., and Zhang, W. (2018). Incentive-aware virtual machine scheduling in cloud computing. *The J. Supercomput.* 74, 3016–3038.. doi: 10.1007/s11227-018-2349-y

Xu, H., Xu, S., Wei, W., and Guo, N. (2023). Fault tolerance and quality of service aware virtual machine scheduling algorithm in cloud data centers. *The J. Supercomput.* 79, 2603–2625. doi: 10.1007/s11227-022-04760-5

Xu, M., Tian, W., and Buyya, R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurr. Comput. Prac. Exp.* 29:4123. doi: 10.1002/cpe.4123

Xu, X., Li, Y., Yuan, Y., Peng, K., Yu, W., Dou, W., et al. (2018). "An energy-aware virtual machine scheduling method for cloudlets in wireless metropolitan area networks," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE.

Xu, X., Zhang, Q., Maneas, S., Sotiriadis, S., Gavan, C., Bessis, N., et al. (2019). VMSAGE: a virtual machine scheduling algorithm based on the gravitational effect for green cloud computing. *Simulation Modelling Prac. Theor.* 93, 87–103. doi: 10.1016/j.simpat.2018.10.006

Xu, X., Zhang, X., Khan, M., Dou, W., Xue, S., Yu, S., et al. (2020). A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. *Future Gener. Comput. Syst.* 105, 789–799. doi: 10.1016/j.future.2017.08.057

Yadav, A. M., and Sharma, S. C. (2023). Cooperative task scheduling secured with blockchain in sustainable mobile edge computing. *Sust. Comput. Inf. Syst.* 37:100843. doi: 10.1016/j.suscom.2022.100843

Yao, F., Yao, Y., Chen, H., Li, T., Lin, M., Zhang, X., et al. (2019). An efficient virtual machine allocation algorithm for parallel and distributed simulation applications. *Concurr. Comput. Prac. Exp.* 31:e5237. doi: 10.1002/cpe.5237

Yu, C., Qin, L., and Zhou, J. (2019). A lock-aware virtual machine scheduling scheme for synchronization performance. *The J. Supercomput.* 75, 20–32. doi: 10.1007/s11227-015-1557-y

Zhan, Z. H., Liu, X. F., Gong, Y. J., Zhang, J., Chung, H. S. H., Li, Y., et al. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Comput. Surv.* 47, 1–33. doi: 10.1145/2788397

Zhao, J., Mhedheb, Y., Tao, J., Jrad, F., Liu, Q., Streit, A., et al. (2014). Using a vision cognitive algorithm to schedule virtual machines. *Int. J. Appl. Mathematics Comput. Sci.* 24, 535–550. doi: 10.2478/amcs-2014-0039

Zhao, Y., Liu, H., Wang, Y., Zhang, Z., and Zuo, D. (2019). Reducing the upfront cost of private clouds with clairvoyant virtual machine placement. *The J. Supercomput.* 75, 340–369. doi: 10.1007/s11227-018-02730-4

Zhou, J., and Yao, X. (2017). Hybrid teaching–learning-based optimization of correlation-aware service composition in cloud manufacturing. *The Int. J. Adv. Manuf. Technol.* 91, 3515–3533. doi: 10.1007/s00170-017-0008-8

Zhu, S. F., Cai, J. H., and Sun, E. L. (2023). Mobile edge computing offloading scheme based on improved multi-objective immune cloning algorithm. *Wireless Netw.* 29, 1737–1750. doi: 10.1007/s11276-022-03157-9

Zhu, T., Shi, T., Li, J., Cai, Z., and Zhou, X. (2018). Task scheduling in deadline-aware mobile edge computing systems. *IEEE Int. Things J.* 6, 4854–4866. doi: 10.1109/JIOT.2018.2874954