JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Comparative Analysis of Scheduling Approaches in Cloud-Fog Computing

[1]Saiyam Varshney, [2]Gur Mauj Saran Srivastava
[1]Research Scholar, [2]Professor
[1]Department of Physics and Computer Science
[1]Dayalbagh Educational Institute, Agra, UP, India

Abstract: With the rapid growth of IoT devices and applications, massive amounts of data are generated and processed to address real-world challenges. Fog computing has emerged as a vital solution, enabling data storage and processing closer to the devices through fog nodes, thereby reducing latency and improving system responsiveness. However, the effectiveness of fog computing environments largely depends on efficient scheduling, which ensures optimal Quality of Service (QoS) by managing tasks, resources, workflows, and jobs while balancing key metrics like latency, execution time, and makespan. In this paper, we present a comprehensive analysis of various scheduling approaches in cloud-fog environments, evaluating their trade-offs and optimization strategies. We also explore the tools used for scheduling, their performance parameters, and the challenges encountered. The insights gained from this study offer valuable guidance for improving scheduling strategies and enhancing the efficiency of cloud-fog environments.

Index Terms **- Cloud and Fog Computing; Internet of things (IoT); Scheduling Problem; Quality of service (QoS).**

I. Introduction

The rapid growth and widespread adoption of Internet of Things (IoT) devices, particularly those integrated with location-aware applications, has resulted in the generation of vast amounts of data. These devices, which are distributed across various environments, are increasingly pushing the limits of traditional cloud computing by sending enormous volumes of data for storage and processing. This shift is creating significant challenges in terms of data storage, computation, and real-time decision-making. Cloud computing has long been the solution for handling large-scale data, but as the number of IoT devices continues to rise, the demand for cloud resources is increasing at an unprecedented rate, leading to potential scalability and latency issues.

According to industry reports, the global cloud computing market is expected to grow by 230%, reaching $623.3 billion by the end of 2023, up from $272 billion in 2018 [1]. This rapid growth is driven in part by the expansion of IoT devices, which are expected to number over 100 billion by 2025, with an economic impact exceeding $11 trillion [2],[3]. As IoT applications become more pervasive, the reliance on cloud infrastructure is straining its capacity to efficiently process and store the enormous amounts of data generated by these devices.

In response to these challenges, fog computing has emerged as a critical solution. Fog computing extends cloud capabilities closer to the edge of the network, bringing data processing and storage closer to the source of data generation—IoT devices [4]. By enabling real-time, localized data processing, fog computing reduces latency, conserves bandwidth, and provides the scalability necessary to meet the demands of an increasingly connected world [5].

However, the efficient management of resources and tasks in a cloud-fog environment remains a significant challenge. Effective scheduling mechanisms are crucial to ensuring that tasks, resources, workflows, and jobs are efficiently managed while maintaining Quality of Service (QoS) [6]. These scheduling mechanisms must balance key metrics such as latency, execution time, makespan, energy consumption, and resource utilization [7].

This paper offers a comprehensive analysis of various scheduling approaches in cloud-fog computing environments, exploring the trade-offs and optimization strategies that help create more efficient systems. We also investigate the tools and techniques used for scheduling, examining their performance parameters and the challenges encountered in managing tasks and resources in dynamic, distributed environments. The goal of this paper is to provide insights that will aid in the development of more effective scheduling solutions for cloud-fog systems, enabling them to better address the growing demands of IoT applications.

The paper is structured as follows: Section 2 presents related work, providing an overview of existing surveys and studies on scheduling approaches in cloud-fog computing. Section 3 describes the scheduling problems and objective services in cloud-fog computing, supported by an in-depth literature review. Section 4 conducts a detailed survey of various scheduling approaches, categorizing and discussing their methodologies. **S**ection 5 provides an analysis and comparison of existing scheduling algorithms, highlighting their strengths, weaknesses, and trade-offs. Finally, Section 6 concludes the paper with a summary of findings and suggestions for future research, followed by references.

II. Related Work

In this section, we present the related works and survey papers based on the scheduling approaches for efficiently accessing the resources in the cloud-fog computing environment. When IoT devices can request services provided by the fog nodes, with each service request having a set of tasks. Scheduling algorithms that find suitable nodes to perform various tasks while meeting the Quality of Services (QoS) requirements and performance metrics reduce the execution time in the cloud-fog environments.

In [8], the author provides a systematic review of the literature on resource management techniques; these techniques are contrasted with each other in terms of crucial factors, including case studies, performance metrics, techniques used, and evaluation tools, and also discusses the benefits and drawbacks in the fog environment.

In [9], the author provides a review of fog computing and its characteristics, threats, and drawbacks and also compares it with various scheduling algorithms.

In [10], the author provides a review of research on task scheduling and computation offloading in cloud, edge, and fog computing environments. They also discuss machine learning techniques for these tasks.

In [11], the author describes the concept of fog computing, along with architecture and its components of fog computing, which are related and distinct from other paradigms in detail Furthermore, they also discuss various resource allocation and scheduling strategies in the fog environment.

In [12], the author provides a survey to analyze task schedules in the form of static and dynamic approaches in fog computing from 2015 to 2018.

In [13], the authors provide a survey on the many scheduling and load-balancing algorithms with their drawbacks process in cloud-fog computing environments.

Consistently, examining previous reviews and survey papers revealed some drawbacks which are as follows:

- Most of the papers don’t consider the latest scheduling algorithms.
- Several papers address the specific scheduling approaches, including resource management [8], [11], [13], task scheduling [12], and workflow scheduling [14], [15].
- Scheduling problems in fog node and cloud-fog computing are not covered in the publications.

Due to the aforementioned factors, we have prepared analysis papers on scheduling problems in cloud-fog computing to address all these vulnerabilities.

The scheduling approach in cloud-fog computing involves efficiently allocating tasks between cloud and fog nodes to optimize performance metrics such as latency, energy consumption, and resource utilization. This is crucial for applications with stringent latency requirements, such as those in the Internet of Things (IoT). Various methodologies have been proposed to address the challenges of task scheduling in these environments, leveraging advanced algorithms and architectures.

III. DESCRIPTION OF SCHEDULING PROBLEMS AND OBJECTIVES SERVICES

In this section, we examine the scheduling challenges and objectives within fog and cloud-fog computing environments, which have become pivotal for modern applications, especially in the Internet of Things (IoT). The hybrid cloud-fog approach merges the power of centralized cloud computing with the flexibility of decentralized fog nodes, enhancing processing efficiency and data management. However, this approach also introduces several complex scheduling issues that must be addressed to achieve optimal performance and deliver high-quality services.

A. Scheduling Problems

The scheduling problem in cloud-fog computing arises from the challenge of efficiently allocating computational tasks across a hybrid infrastructure that combines centralized cloud resources with decentralized fog nodes. As the use of Internet of Things (IoT) devices and real-time applications continues to grow, scheduling becomes increasingly complex due to factors such as resource heterogeneity, dynamic environments, and the diverse requirements of different applications, all of which demand sophisticated scheduling strategies, as shown in Figure 1.

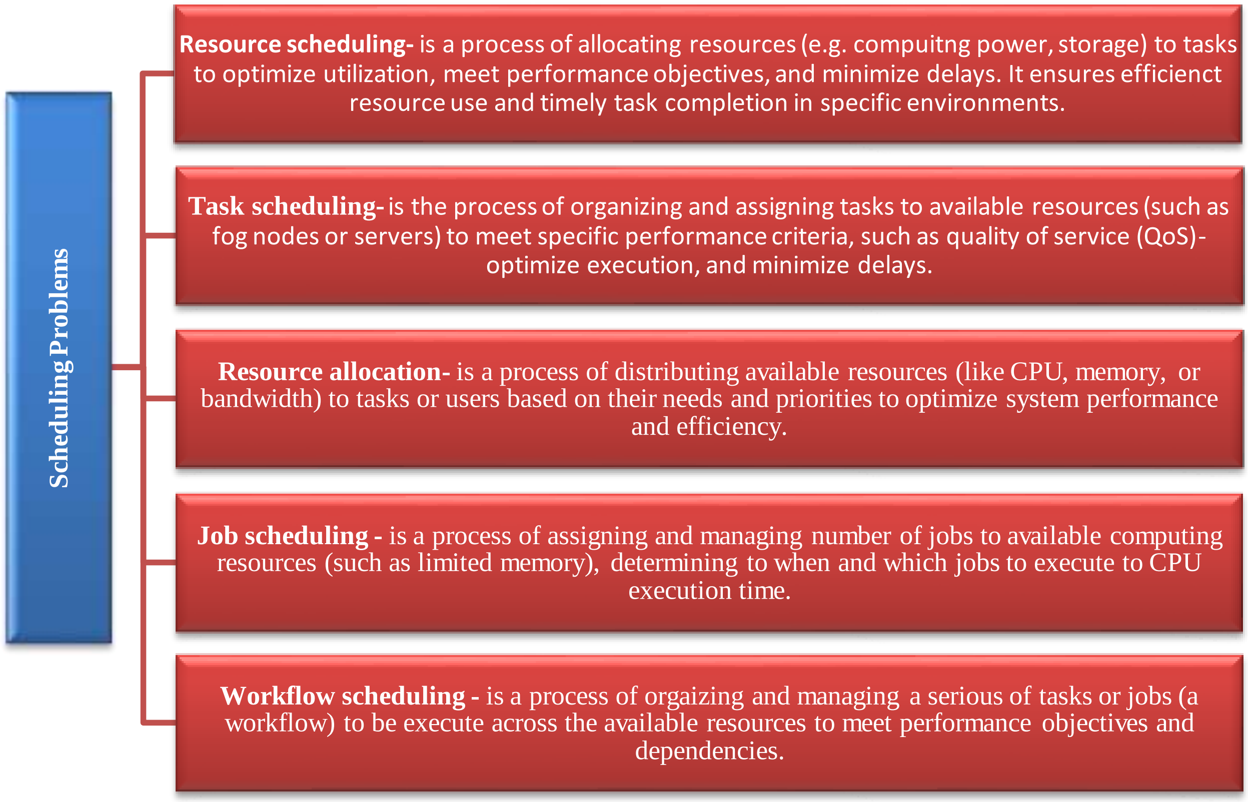

Fig.1: Scheduling problems approach.

B. Scheduling Objectives

The scheduling process involves assigning tasks within workflows to appropriate resources based on specific criteria. These scheduling parameters are crucial for the success of the workflow scheduling problem. The scheduling objectives can be categorized into two groups based on the service approach: service provider and consumer services [32].

Consumer Service:

1. Makespan: Makespan refers to the total time required to complete all tasks in a workflow. It can be seen as the duration from when the user submits the job to the moment the job is completed and its results are generated. Minimizing makespan is essential for improving efficiency and user satisfaction.
2. Budget: The budget represents the financial constraint for utilizing resources. To run a complete workflow, different types of Virtual Machines (VMs) may be employed. The total execution cost, which is the sum of all VMs used, must stay within the user-defined budget.

3. Deadline: For critical applications, there is a strict requirement for completing tasks within a certain time frame. Scheduling must be done with the deadline in mind, ensuring that tasks are completed before the set time limit to meet the application's needs.
4. **Security**: In distributed environments like fog computing, the vast and varied resources make security a major concern. Protecting data and maintaining privacy in fog environments is more complex than in traditional centralized systems due to the distributed nature and potential vulnerabilities of the network.
5. **Cost**: This parameter includes various associated costs such as computing costs, data transmission costs, and storage costs. Efficient scheduling aims to minimize these costs while ensuring optimal resource allocation.

Service Provider

1. **Load Balancing:** Virtual Machines (VMs) are critical resources in the computing environment. During task scheduling, it is common to assign multiple tasks to a VM to run simultaneously. However, this can lead to load imbalances on VMs. Ensuring effective load balancing between resources is essential to improve resource efficiency, which ultimately enhances the overall performance of the scheduling process.
2. **Resource Consumption:** Efficient allocation and consumption of resources are key for service providers. By ensuring that limited resources are fully utilized, service providers can maximize the benefit derived from their available resources. The goal is to optimize resource allocation, ensuring that no resource is underutilized while providing sufficient capacity for users.
3. **Energy Efficiency:** Energy consumption is directly influenced by how processors and resources are used. Inefficient use of processors, such as leaving them idle, results in higher energy consumption. An energy-efficient scheduling process ensures that resources are used optimally, minimizing unnecessary power usage and contributing to the system’s overall sustainability.

IV. SURVEY OF SCHEDULING APPROACHES

In this section, we explore various scheduling approaches employed in cloud-fog computing environments, focusing on task allocation across hybrid infrastructures that integrate centralized cloud resources with decentralized fog nodes. We examine a range of methods designed to optimize performance, energy efficiency, cost, and system effectiveness, categorizing them into resource scheduling, task scheduling, workflow scheduling, and job scheduling. Additionally, we analyze the strategies used to address scheduling challenges from both the consumer's and provider's perspectives, as discussed in the previous sections.

Table 1 provides a summary of the main concepts regarding scheduling algorithm issues, including their achievements, drawbacks, and the simulation tools applied to them. This table helps academicians and researchers compare algorithms based on the approach employed to address a specific scheduling problem, as well as the benefits and drawbacks of each method. As a result, researchers can choose and develop the algorithms that are most relevant to their study.

Table 2 presents a comparison of a wide range of proposed scheduling issues: resource scheduling, task scheduling, resource allocation, job scheduling, and workflow scheduling, classified by problem statements and the performance metrics used in the algorithms. The table also includes a hierarchical comparison of the observed algorithms, along with application case studies and their implementation in cloud or fog environments.

Table 1 A comparison of current research work in the cloud-fog infrastructure.

Paper (s)	Purpose	Toolkit	Achievements	Drawbacks
Saif et al. [17]	A proposed multi-objective task scheduling approach using the Grey Wolf Optimizer algorithm in cloud-fog computing aims to optimize QoS objectives, such as delay and energy consumption, with tasks managed by the fog broker.	Matlab	Verify the effectiveness and reduce the delay and energy consumption compared to another related approach.	Not focused on load balancing, transmission costs, and computing resources using AI approaches.
Azizi et al. [18]	A proposed task scheduling approach using PSG with the multi-start process (PSG-M) optimizes the overall energy consumption of the system by taking into account the deadlines of the tasks.	C++	The proposed task scheduling approach increases the ratio of IoT tasks to their deadline requirements by 1.3x times, reduces the total number of deadline violations by 97.6%, and optimizes the makespan and energy consumption of the fog resource.	Not concerned with improved performance and real-world datasets.
Hoseiny et al. [19]	The proposed algorithms jointly reduce the costs associated with IoT request computation, communication, and latency violations.	Matlab	The proposed algorithms increase the task completion rate by about 99.5% and reduce the overall cost by 15-53% compared to genetic-based algorithms.	Not concerned with energy consumption and real data sets.
Ghobaei-Arani et al. [20]	A proposed Moth-Flame Optimization (MFO) based task scheduling algorithm to minimize the total execution time of tasks to meet the satisfaction of Quality of Service (QoS) requirements of Cyber-Physical System (CPS) applications.	iFogSim	The proposed algorithm provides better results as compared with the three algorithms (PSO, NSGA-II, BLA) by minimizing the total execution time of tasks.	Not concerned with energy consumption and communication costs.
Nguyen et al. [21]	A proposed model is developed to solve a multi-objective optimization problem by determining data processing tasks in a fog-cloud environment, considering three constraints: power consumption, service latency, and monetary cost.	Unknown simulator	A proposed model improves constraints by evaluating PSO, WOA, GA, BLA, and RR metaheuristic models for task scheduling problems in fog cloud environments.	This approach does not consider the real conditions of the fog-cloud systems.
Nikoui et al. [22]	A proposed cost-aware genetic-based (CAG) task scheduling algorithm to improve cost efficiency, performance, and resource allocation in real-time applications with hard deadlines in fog-cloud environments.	iFogSim	A proposed algorithm provides better efficiency in terms of cost and throughput.	Not considering the workflow scheduling under realistic workloads.
Wang et al. [23]	A new task scheduling method (I-FASC) has been proposed to characterize tasks and resources, which includes an improved firework algorithm (I-FA) by introducing a firework burst radius detection mechanism.	Alibaba Cloud Server	This approach optimally minimizes the processing time of tasks and ensures better overall load balancing of the fog appliances.	Not considering the energy consumption of the fog device processing task.
Basir et al. [24]	Resource allocation and performance enhancements for real-time IoT applications, which addresses optimal resource allocation for Cloudlet-IoT networks to maximize overall network performance.	Unknown simulator	A developed Fog-IoT network to provide ε-optimal resource allocation to maximize overall network throughput.	Not consider admission control factor and cache availability on cloudlet-node.
Aburukba et al. [25]	A genetic algorithm (GA) is proposed as a heuristic approach to scheduling IoT requests to achieve the objective of reducing overall latency.	Discrete event simulator	The proposed approach provides improved performance from 21.9% to 46.6% compared to other algorithms and significantly improves the time requests deadline by 31%.	Not addressed to the critical request scheduling, allow preemption and multiple objective functions to optimize the resource usage and reduce latency.
Lee et al. [26]	The proposed algorithm employs reinforcement learning, integrating vehicle movement and parking data from a smart city environment to optimize resource allocation decisions effectively.	Unknown simulator	The VFC resource allocation algorithm provides higher service satisfaction than traditional resource allocation algorithms.	Not concerned with resource constraints, reliance on parked vehicles, simulation-based results, algorithm complexity, and narrow application focus.
Fellir et al. [27]	A proposed multi-agent-based task scheduling algorithm to achieve a balance between application performance and cloud resource utilization performance in a cloud-fog platform.	iFogSim	The proposed algorithm improves resource utilization and performance.	Not concerned with Security and bandwidth obstacles in centralized systems.
Jamil et al. [28]	An efficient job scheduling algorithm (SJF) is proposed to reduce energy consumption, response time, and delay for latency-critical applications.	iFogSim	The proposed algorithm improves the delay by 32% and network usage by 16% as compared to the FCFS algorithm.	Not considered Strave tuples of large lengths
Hussain et al. [29]	An HFSGA approach is proposed to satisfy the task scheduling process to reduce the cost and increase PDST and makespan in the fog-cloud environments.	Matlab	The proposed approach satisfies the necessities of QoS and decreases the costs, increases PDST and Makespan, and is compared with the ACO, PSO, GA, Min-CCV, and Min-V.	Not considering security and load balancing.
Jain et al. [30]	A fog-to-fog offloading method is proposed in the event of an overload to select the most suitable node to maintain the quality of services.	iFogSim	The proposed method is reduced average latency by 65.15% and energy consumption by 67.94%, while 91.07% of total tasks executed successfully.	
Najafizadeh et al. [31]	A multi-objective simulated annealing (MOSA) is proposed for securely allocating tasks on the cloud and fog nodes in terms of	Unknown simulator	The proposed algorithm by using the Goal Programming Approach (GPA) satisfies the multiple objectives.	Not considering User privacy and workflow scheduling of dynamic

	deadline constraints.			parameters.
Demaio et al. [32]	An efficient workflow offloading approach is proposed for fog computing	Monte-Carlo	The proposed single objective algorithm outperforms HEFT by 30% for small task sizes while maintaining the cost and reliability values as compared to MINCOST and MAXREL respectively.	Dynamically changing fog overlays are not considered, including some objectives such as energy consumption and profit.
Ijaz et al. [33]	A novel energy-aware workflow scheduling model is proposed to optimize makespan and energy consumption in fog environments.	Matlab	The proposed model provides less complexity in terms of trade-offs between makespan and energy.	Not considering deadline constraints.
Kaur et al. [34]	A PSW-fog clustering algorithm is proposed for performing scientific workflow applications in a fog-cloud environment	iFogSim	The proposed approach minimizes the time delay, computational cost, and energy consumption of fog nodes.	Not considering the real-world scenario, security of nodes, and
Bisht et al. [35]	The load and Cost Aware Min-Min (LCAMM) algorithm is proposed for workflow applications by taking into account makespan and cost minimization along with the distribution of load among resources.	iFogSim with WorkflowSim	The proposed algorithm gives minimum makespan, less energy consumption, along with load balancing and marginally less cost.	Not considering deadline constraints and response time.
Arshed et al. [36]	A GA-IRACE: Genetic Algorithm-Based Improved Resource Aware Cost-Efficient Scheduler provides the solution to find an efficient scheduling approach for mapping application modules in a cloud fog computing environment.	iFogSim	The proposed algorithm improves the performance by 15-40% in terms of execution time, cost, and bandwidth consumption.	Not consider large corpus applications.
V et al. [37]	An energy-cost-make-aware scheduling algorithm based on a directed acyclic graph (DAG) to represent the graphical dependencies of tasks in a workload.	CloudSim	The proposed algorithm improves the CBTSA by considering both the computation time and the finiteness of cloud resources.	Not focused on heterogeneous IoT devices meeting their QoS requirements by incorporating additional network parameters, such as throughput and average link utilization.
Abohamama et al. [38]	A semi-dynamic real-time task scheduling algorithm is proposed for bag-of-task applications as a permutation-based optimization problem.	Matlab	The proposed algorithm achieves a good balance between the makespan and the total execution cost and reduces the task failure rate compared to the other algorithms.	Not concerned with dynamic scheduling problems in terms of workflow scheduling and load balancing using deep learning algorithms
Husain et al. [39]	A proposed smart scheduling framework improves resource utilization by using the Comparison Attribute Algorithm (CAA) to rank jobs, and the Linear Attribute Summarization Algorithm (LASA) to select accessible CFs with the highest computing capabilities.	Unknown simulator	This proposed framework improves energy consumption and increases the availability of bandwidth with efficient utilization of other sources.	Not focused on the increased task executions by fog devices demanding more resources and the emphasis on efficient resource utilization over the addition of more fog sources.
Huang et al. [40]	The Lyapunov-based PSO algorithm is proposed to reduce energy consumption in the balanced distribution of transmission power and computing resources through the control of channel resources.	EdgeCloudSim	This proposed algorithm provides better results than PSO and greedy algorithms in the performance index of energy consumption required to complete the task.	Not focusing on task decomposition through the use of a directed acyclic graph (DAG) to better handle subtasks with sequential relationships.
Shukla and Pandey [41]	A proposed MOTOR algorithm to enhance workflow execution in heterogeneous fog-cloud infrastructures by tackling resource scheduling, task offloading, and workflow optimization challenges.	iFogSim with WorkflowSim	The proposed algorithm optimizes task offloading and resource scheduling, ensuring efficiency in makespan, cost, resource utilization, and energy consumption, with an optimal number of VMs for FL and CL collaboration.	Not focused on task-resource mapping for communication-intensive applications but on efficiency within deadline and cost constraints
Thakur et al. [42]	A proposed task scheduling algorithm using hybrid fuzzy logic improves the semi-greedy approach, enabling the algorithm to make smarter decisions and better adapt to the dynamic and uncertain conditions of the fog environment.	C++	This algorithm provides better performance in terms of reducing makespan and energy consumption (EC), as well as increasing the percentage of deadline-satisfied tasks compared to PSG and PSG-M.	Not concerned with improving the scheduling of dependent IoT tasks or evaluating their performance using real-world datasets to ensure practical applicability.
Wu et al. [43]	A multi-objective Estimation of Distribution Algorithm (EDA) is proposed to optimize task offloading in IoT–fog–cloud systems, using a fuzzy logic strategy to handle uncertainty, enhance robustness, and reduce complexity through application clustering.	C++	The algorithm outperformed classic heuristic solutions in 88.3% of benchmark cases and dominated two state-of-the-art multi-objective algorithms in 92.7% and 94.4% of cases while achieving significant resource savings and robust scalability.	Not focused on integrating offline training with an online model to reduce algorithm time complexity, but rather on developing a prototype system for design validation.
Varshney and Srivastava [44]	A proposed ACO-based workflow scheduling in fog-cloud computing environments that efficiently allocates task loads to fog and cloud resources, thereby enhancing execution time and reducing costs within the distributed environment.	iFogSim with WorkflowSim	The algorithm provides better results of the proposed approach using well-known scientific workflows suggesting that cloud-fog integration for workflow scheduling improves processing performance while reducing cost.	Not concerned with energy consumption and other QoS parameters for real-time and benchmark datasets.

Khaledian [45]	A hybrid PSO-SA algorithm is proposed to optimize energy consumption and makespan by prioritizing tasks and managing task allocation in fog-cloud environments.	Matlab	The proposed algorithm improves energy consumption by 5% and makespan by 9% compared to the baseline algorithm (IKH-EFT).	Not addressing node failures or the increased complexity arising from task deadlines, priorities, and workflow maintenance.
Mahboubeh Afzali et al. [46]	A proposed IBPSO algorithm optimizes IoT resource allocation in hybrid fog-cloud systems, reducing latency and ensuring load balancing for smart city applications.	Unknown simulator	The proposed algorithm reduces latency by up to 28% and improves missed deadlines, runtime, and load balancing by 11%-22% compared to other methods, with a 45% better runtime efficiency than BGA, BPSO, and BGWO.	Not concerned with other applications
Yadav et el. [47]	A proposed hybrid BH-FWA algorithm is to optimize task scheduling in fog computing networks by combining HEFT and FWA techniques, aiming to minimize makespan and cost, thus improving QoS for IoT-enabled smart applications.	iFogSim	The proposed algorithm improves job scheduling compared to HEFT, RC-GA, IMFWA, and GA-PSO, achieving 12% to 37% better cost and makespan, with superior throughput and robustness.	Not concerned with multi-objective variants, varying bandwidths, and fault tolerance.
Mukherjee [48]	A proposed task scheduling scheme provides the optimal task scheduling scheme that uniformly distributes tasks among fog nodes based on Quality of Service (QoS) requirements	iFogSim	The proposed algorithm result provides efficient task distribution and reduces energy consumption by 1.26% compared to a state-of-the-art competitor.	Not concerned with the lack of real-world validation or the limited evaluation, being compared against only one competitor.
Qi et al. [49]	The proposed multi-objective scheduling algorithm, IPAQ, enhances fog computing by optimizing resource allocation, response times, task prioritization by time sensitivity, and real-time handling of large task volumes under multi-objective conditions.	iFogSim	The proposed algorithm achieves superior performance in managing large volumes of tasks within fog computing environments, delivering significant improvements in response times, system efficiency, and resource allocation when compared to existing algorithms.	Not focus on energy- and cost-efficient task scheduling strategies and flexible resource management in large-scale networks.
Yu et al. [50]	Proposed a heuristic-based Data-Locality Aware Job Scheduling in Fog-Cloud (DLSFC) algorithm to optimize the scheduling of data-intensive tasks in fog and cloud-based IoT	CloudSim	The proposed algorithm achieves solutions within 85% of the optimal results, ensuring effective task scheduling and resource utilization.	This approach lacks task prioritization, data migration rules, and criteria such as energy consumption and cost.
Kaur et al. [51]	The paper presents "EcoFogLoad Architecture" and "EEWO" algorithms to optimize fog computing resource allocation, enhancing workload balance, and reducing latency, and energy consumption for IoT applications.	iFogSim	The proposed algorithm optimizes fog layer resources by reducing cost, time delay, and energy consumption, thereby lowering latency and enhancing service quality.	Not concerned with addressing server overload or implementing load balancing in fog-cloud environments, despite emphasizing energy efficiency.
Karpe & SH [52]	A proposed novel Coati Integrated Beluga Whale Optimization (CI-BWO) strategy to optimize resource allocation by considering multiple constraints such as resource utilization, service response rate, and communication cost.	Unknown simulator	The proposed strategy significantly reduces migration costs compared to traditional methods, achieving a migration cost of 1.287 for 200 tasks, which is lower than existing models.	Not concerned with inefficiencies and high migration costs of traditional resource allocation in fog computing
Kuppusamy et al. [53]	Proposed a novel job scheduling algorithm using Dynamic Opposition Learning-based Social Spider Optimization (DOLSSO) to optimize job scheduling by minimizing energy consumption, reducing makespan, and ensuring efficient resource allocation for the fog system.	iFogSim	The proposed algorithm result achieves 10% - 15% better CPU utilization and 5%-10% less energy consumption than the other techniques.	Not concerned with critical, less critical, and non-critical tasks, and not fully addressing dynamic healthcare workloads or large-scale deployments.
Choppara and Mangalampalli [54]	A proposed DRLMOTS for efficient task scheduling in cloud-fog environments to optimize the makespan and energy consumption.	SimPy (Python based discrete event)	The proposed algorithm results in achieving up to 26.80% reduction in makespan and 39.60% decrease in energy consumption, while enhancing fault tolerance by up to 221.89%.	Not focused on the OpenStack environment to evaluate the effectiveness of this algorithm's performance.
Khan et al. [55]	A proposed workload-aware framework for resource allocation and load balancing optimizes ECG processing in IoHT applications, reducing delays, energy use, and network consumption while enhancing throughput in fog computing.	iFogSim	The proposed technique reduces delay by 45%, energy consumption by 37%, and network bandwidth usage by 25% compared to existing studies.	Not focused on prioritizing critical tasks in dynamic healthcare workloads to optimize fog resources and minimize delay, energy, and response time, nor addressing heuristic and meta-heuristic challenges.
Marwa et al. [56]	A proposed Fuzzy-Cone approach uses a multi-agent system to optimize workflow scheduling in Fog-Cloud environments by negotiating conflicting constraints between client and supplier agents, leveraging fuzzy inference to optimize time and cost.	JADE	The proposed approach achieves better results by reducing compilation time by 37% and increasing cost by 6% compared to GA, PSO, and MASGA.	Not focused on comparing the Fuzzy-Cone approach with existing methods, developing a Pareto-based multi-objective optimization for scheduling, or exploring cooperation strategies for a balanced scheduling solution.
Potu et al. [57]	A proposed extended particle swarm optimization (EPSO) algorithm optimizes task scheduling in cloud-fog environments, enhancing resource efficiency and reducing	iFogSim	The proposed algorithm achieved a makespan of 337.50 s and a total cost of 1391.58 G$, outperforming MPSO in cost efficiency and processing	Not focused on developing a strategy for balancing delay and energy consumption using parallel meta-heuristic

	task completion time.		performance, though it increased energy consumption at the fog node level.	algorithms.
Baskar et al. [58]	A proposed Hybrid Prairie Dog and Dwarf Mongoose Optimization Algorithm-based Resource Scheduling (HPDDMOARS) technique aims to optimize resource scheduling in IoT-fog environments.	Yet Another Fog Simulator (YAFS)	The proposed scheme achieved a 22.18% reduction in energy consumption, a 24.98% decrease in makespan, and an 18.64% cost reduction across varying numbers of IoT applications, outperforming baseline metaheuristic deployment approaches.	Not focused on managing high computational complexity and scalability challenges when handling large numbers of IoT tasks in resource-constrained environments.

Table 2 A detailed summary of the used in this study

Approaches	Reference	Algorithms	Metrics	Environments	Case study	Result Compared
Resource Scheduling	[25], 2020	GA	• Latency • Average Delays • Processing power • number of resources	Cloud-fog	Mobile applications	WFQ, PSQ, and RR
	[57], 2021	EPSO	• Makespan, • Cost, • resource utilization, • response time, and • latency	Fog	General	TCaS, ideal PSO, MPSO, and BLA
	[39], 2022	Smart MF	• Energy consumption • Resource utilization	Cloud-fog	Smart Cities Applications	CAA and LASA
	[40], 2022	LPSO	• Energy consumption • Deadline miss ratio	Fog	General	PSO and Greedy
	[41], 2023	MOTOR	• Makespan • Energy consumption • Cost	Cloud-fog	Workflow application	ACO, HPSOGWO, and MAA
	[58], 2025	HPDDMOARS	• Makespan • Energy • Service Cost	Fog	General	FP- MPSOCA, CSAMMAPA, EWOAMHM and NSGAMAP
Task Scheduling	[21], 2020	metaheuristic algorithms	• Power consumption • Service latency • Monetary cost	Cloud-fog	General	GA, PSO, WOA, BLA, and Round-Robin
	[23], 2020	I-FASC and I-FA	• execution time • load in a short time by two sets of experiments	Cloud-fog	Mobile Applications (actual)	DFGA, Rank-ACO, and FA
	[20], 2020	TS-MFO	• Makespan	Fog	CPS applications	PSO, NSGA-II, and BLA
	[22], 2020	CAG	• Success rate • Cost per Instruction • Cost	Cloud-fog	Real-time applications	RR and MRT
	[27], 2020	Multi-Agent	• energy consumption, • Cost.	Cloud-fog	General	FCFS
	[19], 2021	Min-V	• PDST • Makespan • Violation cost • Computation cost • Communication cost • total cost	Cloud-fog	General	RR, Random, TCaS, Min-CCV
	[29], 2022	HFSGA	• Makespan • Communication cost, • Computational cost, • Deadline violation cost, • PDST	Cloud-fog	Large scale applications	ACO, PSO, GA, Min-CCV, Min-V, and RR
	[31], 2022	MOSA	• Service delay time • Access level control • Service cost • Deadline	Cloud-fog	Large scale applications	MOPSO, MOTS, and MOMF
	[38], 2022	IGA-POP	• makespan, • execution cost, • fitness value, • failure rate, and • average delay time	Cloud-fog	Large scale applications	FF, BF, GA, and BLA
	[18], 2022	PSG-M	• makespan • Deadline Satisfaction Deadline violation	Fog	General	FCFS, EDF, GfE, Detour, PSG,
	[47], 2022	BH-FWA	• Makespan • Cost	Fog	General	HEFT, RC-GA, IMFWA, and hybrid GA-PSO
	[17], 2023	MGWO	• delay, and • energy consumption	Cloud-fog	General	cloud-fog cooperation, NGSA-II, and MOPSO
	[42], 2024	FuzzyPSG-M	• Makespan • Energy consumption • Deadline Satisfaction	Fog	General	FCFS, EDF, GfE, Detour, PSG, PSG-M

	[48], 2024	Multilayer fog-based loT system architecture	• Delay, • Energy Consumption, • Fog Node Utilization	Fog	General	EM-MOO
	[54], 2025	DRLMOTS	• Makespan • energy consumption • fault tolerance	Cloud-fog	Real-time applications (Google Jobs)	CNN, LSTM, and GGCN
	[49], 2025	IPAQ	• makespan • Task Time Sensitivity, • Waiting Time, and • response times	Fog	Real-time applications	PSO, GA, MQP, MQF, FCFS, and SJF
Resource allocation	[43], 2021	MOEA/D	• Latency • Resource allocation, and • Scalability	Fog	Real-time applications	Fuzzy EDA (fEDA) and FOHEFT
	[55], 2022	DWTSA	• Loop Delay, • Energy Consumption, • Network Consumption, • Average Throughput	Fog	Healthcare	FCFS and SJF
	[37], 2022	ECBTSA-IRA	• makespan • cost • workload delay	Cloud-fog	General	CBTSA, ECBTSA
	[46], 2023	IBPSO	• latency, • missed deadline requests, • run time, and • load balancing	Cloud-fog	General	BGA, BPSO, BGWO, and ranked-based resource allocation methods
	[51], 2024	EEWO	• cost, • time delay and • energy consumption	Fog	Workflow applications	Tabu Search and GWO
	[52], 2024	CI-BWO	• resource utilization, • service response rate, • makespan, • cost	Fog	General	Traditional RA
Job Scheduling	[28], 2020	SJF	• energy consumption, • response time, • latency, and • network usage	Fog	Healthcare	FCFS
	[36], 2022	GA-IRACE	• Execution Time • Monetary Cost • Bandwidth	Cloud-fog	Real-time Applications	RACE, Random Solution, and MMA
	[53], 2022	DOLSSO	• Makespan, • Energy consumption, • Resource allocation	Cloud-fog	General	SSO, WOA, GWO, SSA, and GOA
	[50], 2025	DLSFC	• Response time, • Energy Consumption • Network Lifetime and Traffic Load	Cloud-fog	Real-time Applications	LP_Migration and LP_Locality
Workflow Scheduling	[32], 2020	MOWO	• Response time, • Reliability, and • Financial cost	Cloud-fog	Large-scale (Real-world) applications	HEFT and PEFT
	[33], 2021	EM-MOO	• Makespan, and • Energy consumption	Cloud-Fog	Workflow applications	EM-MCC, MOPT, and PEFT
	[34], 2022	PSW-Cluster	• Time delay, • Cost, and • Energy consumption	Fog	Workflow applications	ABC, Tabu search, ACO, GWO, and Tabu-GWO-ACO
	[35], 2022	Improved min-min	• cost, • makespan, • energy, and • load balancing	Cloud-fog-edge	Workflow applications	min-min and ELBMM
	[56], 2023	Fuzzy-Cone	• makespan, • cost	Cloud-fog	Workflow scheduling	GA, PSO, and MASGA
	[44], 2024	ACO	• makespan, • cost	Cloud-fog-edge	Workflow applications	RR, SJF, and FCFS
	[45], 2024	PSO-SA	• makespan • energy	Cloud-fog	Workflow application	baseline algorithm (IKH-EFT)

V. Analysis And Comparison

In this section, we discuss and compare the existing literature on scheduling approaches within fog and cloud-fog computing environments. The statistical analysis highlights several critical aspects, including the algorithms, evolutionary environments, tools, and metrics used in these studies, providing valuable insights into their methodologies and applications.

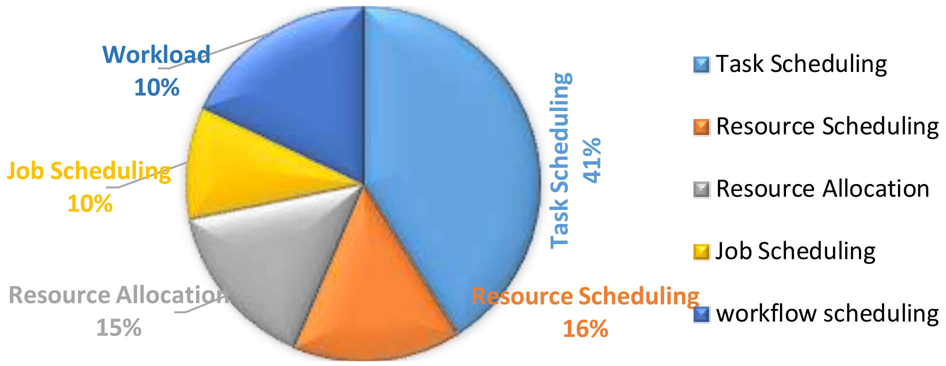

Figure 2 Scheduling problems solved by these approaches

Figure 2 shows a statistical comparison of scheduling problems in the cloud-fog environment, highlighting the significance of the taxonomy outlined in the previous sections. The analysis reveals that task scheduling accounts for 41% of the literature focusing on this area, followed by resource scheduling at 16%, resource allocation at 15%, while job scheduling accounts for 10%, and finally workflow scheduling is the focus of 10% of the research reviewed.

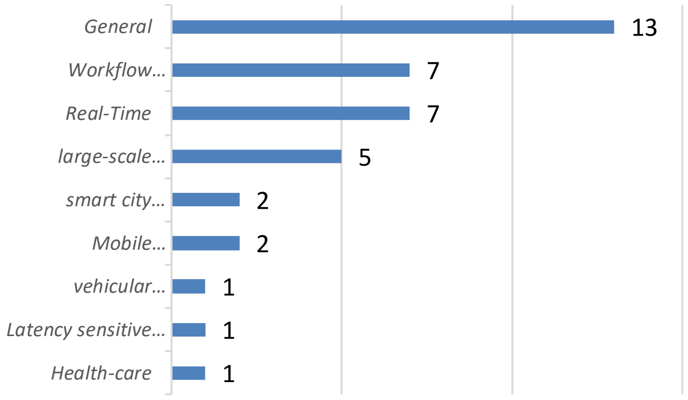

Figure 1 used case studies in various research

As shown in Figure 3, we looked at several case studies on scheduling approaches in cloud-fog and fog paradigms. General applications came out on top, with 15 studies, followed by large-scale applications (11) and mobile applications (7). Industrial and smart city applications had 5 studies each, while workflow applications had 4 and surveillance applications 3. A few areas, like face recognition, healthcare, and latency-sensitive applications, were less studied, with only 1-2 studies each. These scheduling approaches focus on improving IoT systems by cutting down delays, speeding up response times, and boosting accuracy.

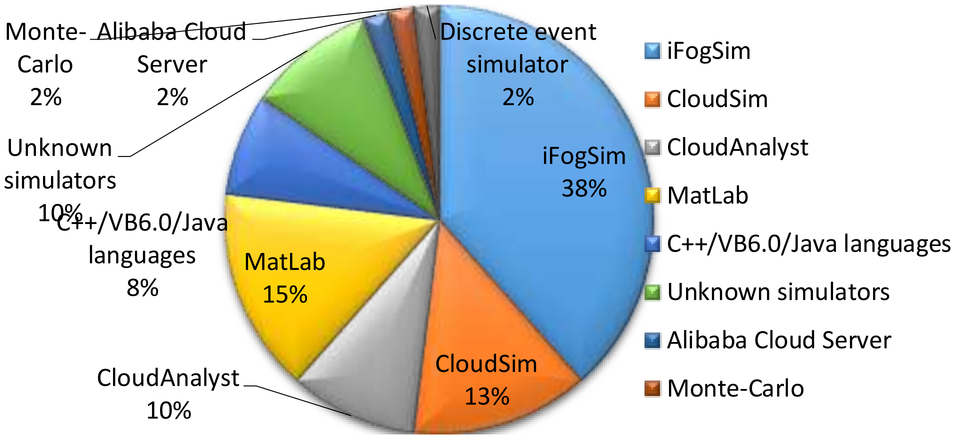

Figure 4 Evaluation tools used in this study

Figure 4 shows the evaluation toolkit used or developed by several researchers in our literature work, with iFogSim used in 39% of the simulation studies. CloudSim tool is used in 13% of the studies. Matlab tool is used in around 15% of the evaluation studies. While remaining studies are based on CloudAnalyst simulators, Monte-Carlo, programming languages e.g. C++/VB6.0/Java, Alibaba cloud servers, and unknown simulators.

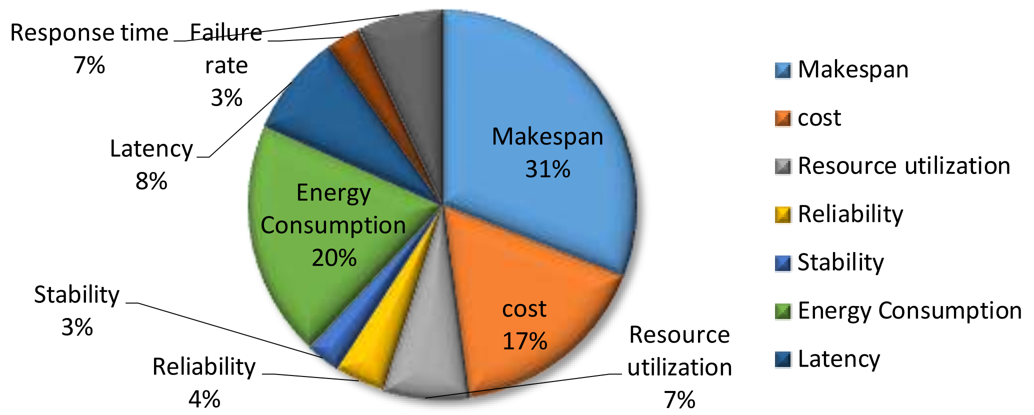

Figure 5 Performance metrics used in scheduling algorithms.

As shown in Figure 5, the literature on scheduling algorithms highlights makespan as the most commonly used performance metric, accounting for 31%, followed by energy consumption and cost optimization at 17% each. Latency is considered in 8% of studies, while resource utilization and response time each make up 7%. Reliability appears in 4%, and less frequently considered metrics, such as failure rate, allocated memory, and stability, are found in only 3% of the studies, indicating potential areas for further research.

VI. Conclusions

In this paper, we review and analyze various scheduling approaches used in both fog and cloud-fog computing environments. We explore different algorithms, highlighting their pros and cons, and assess how they contribute to efficient operations in these settings. We also compare key research studies on scheduling problems, looking at factors like evaluation criteria, successes, and limitations. We recognize that there are still several critical challenges to address in fog and cloud-fog computing. To help overcome these, we suggest future research directions focused on key areas like heterogeneity, security, power consumption, load balancing, financial cost, and response time—issues that have been underexplored in current studies. By combining different approaches and considering these key performance factors, we can significantly improve the effectiveness of scheduling algorithms in both fog and cloud-fog computing environments.

References

[1] Nick Galov, “25 Must-Know Cloud Computing Statistics in 2022,” 2022. [Online]. Available: https://webtribunal.net/blog/cloud-computing-statistics/

[2] M. I. Bala and M. A. Chishti, “Offloading in Cloud and Fog Hybrid Infrastructure Using iFogSim,” in 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2020, pp. 421–426.

[3] A. K. Majumdar, Optical Wireless Communications for Broadband Global Internet Connectivity. Elsevier, 2019.

[4] Cisco, “Fog Computing and the Internet of Things.” 2015.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC ’12, 2012, p. 13.

[6] R. A. C. da Silva and N. L. S. da Fonseca, “On the Location of Fog Nodes in Fog-Cloud Infrastructures,” Sensors, vol. 19, no. 11, p. 2445, May 2019.

[7] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, “Fog of Everything: Energy-Efficient Networked Computing Architectures, Research Challenges, and a Case Study,” IEEE Access, vol. 5, pp. 9882–9910, 2017.

[8] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, “Resource Management Approaches in Fog Computing: a Comprehensive Review,” Journal of Grid Computing, vol. 18, no. 1. Journal of Grid Computing, pp. 1–42, Mar. 06, 2020.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2016.

[10] D. Rahbari and M. Nickray, “Computation Offloading and Scheduling in Edge-Fog Cloud Computing,” J. Electron. Inf. Syst., vol. 1, no. 1, pp. 24–34, 2019.

[11] A. Yousefpour et al., “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” J. Syst. Archit., vol. 98, no. December 2018, pp. 289–330, 2019.

[12] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, “aTask scheduling approaches in fog computing: A survey,” Trans. Emerg. Telecommun. Technol., vol. 33, no. 3, p. e3792, Mar. 2022.

[13] M. M. Mon and M. A. Khine, “Scheduling and Load Balancing in Cloud-Fog Computing using Swarm Optimization Techniques : A Survey,” Sched. Load Balanc. Cloud-Fog Comput. using Swarm Optim. Tech. A Surv. 3 . Overv. Sched. Load, pp. 8–14, 2019.

[14] M. Hosseinzadeh et al., “Task Scheduling Mechanisms for Fog Computing: A Systematic Survey,” IEEE Access, vol. PP, no. May, pp. 1–1, 2023.

[15] A. Jain and R. Rajak, “A systematic review of workflow scheduling techniques in a fog environment,” Int. J. Exp. Res. Rev., vol. 30, pp. 100–108, Apr. 2023.

[16] R. Bouabdallah and F. Fakhfakh, "Workflow Scheduling in Cloud–Fog Computing Environments: A Systematic Literature Review," Concurrency and Computation: Practice and Experience. John Wiley & Sons, Ltd, p. e8304, 2024.

[17] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, "Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing," IEEE Access, vol. 11, no. January, pp. 20635–20646, 2023.

[18] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," J. Netw. Comput. Appl., vol. 201, May 2022.

[19] F. Hoseiny, S. Azizi, M. Shojafar, and R. Tafazolli, "Joint QoS-Aware and Cost-efficient Task Scheduling for Fog-cloud Resources in a Volunteer Computing System," ACM Trans. Internet Technol., vol. 21, no. 4, 2021.

[20] M. Ghobaei-Arani, A. Souri, F. Safara, and M. Norouzi, "An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing," Trans. Emerg. Telecommun. Technol., vol. 31, no. 2, p. e3770, Feb. 2020.

[21] T. Nguyen, K. Doan, G. Nguyen, and B. M. Nguyen, "Modeling Multi-constrained Fog-cloud Environment for Task Scheduling Problem," in 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Nov. 2020, pp. 1–10.

[22] T. S. Nikoui, A. Balador, A. M. Rahmani, and Z. Bakhshi, "Cost-Aware Task Scheduling in Fog-Cloud Environment," in 2020 CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST), Jun. 2020, pp. 1–8.

[23] S. Wang, T. Zhao, and S. Pang, "Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing," IEEE Access, vol. 8, pp. 32385–32394, 2020.

[24] R. Basir, S. Qaisar, M. Ali, H. Pervaiz, M. Naeem, and M. A. Imran, "Resource Allocation and Throughput Maximization for IoT Real-time Applications," in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), May 2020, vol. 2020-May, pp. 1–5.

[25] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing," Futur. Gener. Comput. Syst., vol. 111, pp. 539–551, Oct. 2020.

[26] S. S. Lee and S. Lee, "Resource Allocation for Vehicular Fog Computing Using Reinforcement Learning Combined with Heuristic Information," IEEE Internet Things J., vol. 7, no. 10, pp. 10450–10464, 2020.

[27] F. Fellir, A. El Attar, K. Nafil, and L. Chung, "A multi-Agent based model for task scheduling in cloud-fog computing platform," in 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Feb. 2020, no. February, pp. 377–382.

[28] B. Jamil, M. Shojafar, I. Ahmed, A. Ullah, K. Munir, and H. Ijaz, "A job scheduling algorithm for delay and performance optimization in fog computing," Concurr. Comput. Pract. Exp., vol. 32, no. 7, p. e5581, Apr. 2020.

[29] S. M. Hussain and G. R. Begh, "Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment," J. Comput. Sci., vol. 64, p. 101828, Oct. 2022.

[30] V. Jain and B. Kumar, "A Trusted Resource Allocation Scheme in Fog Environment to Satisfy High Network Demand," Arab. J. Sci. Eng., vol. 48, no. 8, pp. 1–18, Oct. 2022.

[31] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective Task Scheduling in cloud-fog computing using goal programming approach," Cluster Comput., vol. 25, no. 1, pp. 141–165, Feb. 2022.

[32] V. De Maio and D. Kimovski, "Multi-objective scheduling of extreme data scientific workflows in Fog," Futur. Gener. Comput. Syst., vol. 106, pp. 171–184, May 2020.

[33] S. Ijaz, E. U. Munir, S. G. Ahmad, M. M. Rafique, and O. F. Rana, "Energy-makespan optimization of workflow scheduling in fog–cloud computing," Computing, vol. 103, no. 9, pp. 2033–2059, Sep. 2021.

[34] M. Kaur and R. Aron, "An Energy-Efficient Load Balancing Approach for Scientific Workflows in Fog Computing," Wirel. Pers. Commun., vol. 125, no. 4, pp. 3549–3573, Aug. 2022.

[35] J. Bisht and V. S. Vampugani, "Load and Cost-Aware Min-Min Workflow Scheduling Algorithm for Heterogeneous Resources in Fog, Cloud, and Edge Scenarios," Int. J. Cloud Appl. Comput., vol. 12, no. 1, pp. 1–20, Nov. 2022.

[36] J. U. Arshed, M. Ahmed, T. Muhammad, M. Afzal, M. Arif, and B. Bazezew, "GA-IRACE: Genetic Algorithm-Based Improved Resource Aware Cost-Efficient Scheduler for Cloud Fog Computing Environment," Wirel. Commun. Mob. Comput., vol. 2022, 2022.

[37] S. V, P. M, and M. K. P, "Energy-Efficient Task Scheduling and Resource Allocation for Improving the Performance of a Cloud–Fog Environment," Symmetry (Basel)., vol. 14, no. 11, p. 2340, Nov. 2022.

[38] A. S. Abohamama, A. El-Ghamry, and E. Hamouda, "Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment," J. Netw. Syst. Manag., vol. 30, no. 4, p. 54, Oct. 2022.

[39] B. H. Husain and S. Askar, "Smart Resource Scheduling Model in Fog Computing," in 2022 8th International Engineering Conference on Sustainable Technology and Development (IEC), Feb. 2022, pp. 96–101.

[40] C. Huang, H. Wang, L. Zeng, and T. Li, "Resource Scheduling and Energy Consumption Optimization Based on Lyapunov Optimization in Fog Computing," Sensors, vol. 22, no. 9, 2022.

[41] P. Shukla and S. Pandey, "MOTORS: multi-objective task offloading and resource scheduling algorithm for heterogeneous fog-cloud computing scenario," J. Supercomput., vol. 80, no. 15, pp. 22315–22361, Oct. 2024.

[42] R. Thakur, G. Sikka, U. Bansal, J. Giri, and S. Mallik, "Deadline-aware and energy efficient IoT task scheduling using fuzzy logic in fog computing," Multimed. Tools Appl., Jun. 2024.

[43] C. ge Wu, W. Li, L. Wang, and A. Y. Zomaya, "An evolutionary fuzzy scheduler for multi-objective resource allocation in fog computing," Futur. Gener. Comput. Syst., vol. 117, pp. 498–509, 2021.

[44] S. Varshney and G. M. S. Srivastava, "Efficient Workflow Scheduling in Fog-Cloud Environments using Ant Colony Optimization," in 2024 Second International Conference on Data Science and Information System (ICDSIS), May 2024, pp. 1–5.

[45] N. Khaledian, K. Khamforoosh, R. Akraminejad, L. Abualigah, and D. Javaheri, "An energy-efficient and deadline-aware workflow scheduling algorithm in the fog and cloud environment," Computing, vol. 106, no. 1, pp. 109–137, Jan. 2024.

[46] M. Afzali, A. Mohammad Vali Samani, and H. R. Naji, "An efficient resource allocation of IoT requests in hybrid fog–cloud environment," J. Supercomput., vol. 80, no. 4, pp. 4600–4624, Mar. 2024.

[47] A. M. Yadav, K. N. Tripathi, and S. C. Sharma, "A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm," J. Supercomput., vol. 78, no. 3, pp. 4236–4260, 2022.

[48] S. Mukherjee, J. Sengupta, and S. Das Bit, "An Efficient Multi-Objective Task Scheduling in Fog-Cloud Environment," in 2024 IEEE Region 10 Symposium (TENSYMP), Sep. 2024, pp. 1–6.

[49] M. Qi, X. Wu, K. Li, and F. Yang, "IPAQ: a multi-objective global optimal and time-aware task scheduling algorithm for fog computing environments," J. Supercomput., vol. 81, no. 2, p. 377, Jan. 2025.

[50] X. Yu, M. Zhu, M. Zhu, X. Zhou, L. Long, and M. Khodaparast, "Location-aware job scheduling for IoT systems using cloud and fog," Alexandria Eng. J., vol. 110, pp. 346–362, Jan. 2025.

[51] M. Kaur, R. Aron, and S. Seth, "Optimizing Resource Allocation for Energy Efficiency in Fog Cloud Computing Environments," in 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT), Apr. 2024, pp. 538–542.

[52] S. P. Karpe and B. SH, "Resource prioritization and allocation in fog computing using hybrid optimization," Web Intell., vol. Preprint, no. Preprint, pp. 1–22, Aug. 2024.

[53] P. Kuppusamy et al., "Job scheduling problem in fog-cloud-based environment using reinforced social spider optimization," J. Cloud Comput., vol. 11, no. 1, 2022.

[54] P. Choppara and S. Mangalampalli, "An efficient deep reinforcement learning based task scheduler in cloud-fog environment," Cluster Comput., vol. 28, no. 1, pp. 1–26, 2025.

[55] S. Khan, I. Ali Shah, N. Tairan, H. Shah, and M. Faisal Nadeem, "Optimal Resource Allocation in Fog Computing for Healthcare Applications," Comput. Mater. Contin., vol. 71, no. 3, pp. 6147–6163, 2022.

[56] M. Marwa, J. E. Hajlaoui, Y. Sonia, M. N. Omri, and C. Rachid, "Multi-agent system-based fuzzy constraints offer negotiation of workflow scheduling in Fog-Cloud environment," Computing, vol. 105, no. 7, pp. 1361–1393, Jul. 2023.

[57] N. Potu, C. Jatoth, and P. Parvataneni, "Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments," Concurr. Comput. Pract. Exp., vol. 33, no. 23, p. e6163, Dec. 2021.

[58] R. Baskar, E. Mohanraj, M. Saradha, and R. Monika, "Hybrid Prairie Dog and Dwarf Mongoose optimization algorithm-based application placement and resource scheduling technique for fog computing environment," Sci. Rep., vol. 15, no. 1, p. 1240, Jan. 2025.