Task Scheduling in Fog Computing – Classification, Review, Challenges and Future Directions

Deafallah Alsadie
dbsadie@uqu.edu.sa
Department of Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia

Abstract
With the advancement in the Internet of things Technology (IoT) cloud computing, billions of physical devices have been interconnected for sharing and collecting data in different applications. Despite many advancements, some latency - specific application in the real world is not feasible due to existing constraints of IoT devices and distance between cloud and IoT devices. In order to address issues of latency sensitive applications, fog computing has been developed that involves the availability of computing and storage resources at the edge of the network near the IoT devices. However, fog computing suffers from many limitations such as heterogeneity, storage capabilities, processing capability, memory limitations etc. Therefore, it requires an adequate task scheduling method for utilizing computing resources optimally at the fog layer. This work presents a comprehensive review of different task scheduling methods in fog computing. It analyses different task scheduling methods developed for a fog computing environment in multiple dimensions and compares them to highlight the advantages and disadvantages of methods. Finally, it presents promising research directions for fellow researchers in the fog computing environment.

Keywords: *Fog computing, Task scheduling, Job scheduling, Resource provisioning.*

1. Introduction

Cloud computing has resulted in quick implementation of different business policies, sharing of computing resources and business information systems in different factors such as education and industry [1,2]. It uses the virtualization concepts to allocate shared resources, implement bandwidth calculating allocated resources, and load balancing in networking environment through Internet [3,4].

However, recent development in the Internet of things (IoT) has produced a massive amount of data using different sensor devices [5]. The cloud computing environment provides processing and storage services for handling the needs of IoT. However, for latency specific applications, delay in transferring data from cloud to device and vice versa is not acceptable [6]. Transferring a massive amount of data to the cloud for further processing also requires a large amount of network bandwidth that directly impacts scalability.

Fog computing delivers a distributed model analogous to cloud computing services to the network near IoT. It has attracted significant consideration from academicians and industrialists in recent times due to its potential benefits such as real time processing, quick scalability, and location awareness [7]. Figure 1 presents fog and cloud computing environment architecture consisting of three different layers. The topmost layer is the cloud layer. The cloud layer concept of hire and computing services and storage devices.

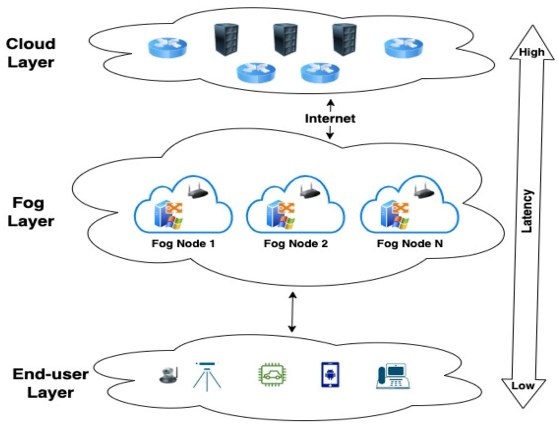

FIGURE 1. CLOUD/FOG COMPUTING ARCHITECTURE

The middle layer is the fog computing layer, which consists of routers, switches, and gateways. These devices offer IoT services at the network edge instead of transferring everything to the cloud. The last layer of the architecture consists of end - user devices producing considerable data.
Several critical issues have been identified regarding fog and cloud architecture. The most critical issues include allocating computing resources, balancing workload, establishing security, achieving optimization of power consumption, fault tolerance and task scheduling. Task scheduling is one of the most critical issues in the fog and cloud computing environment. The task is considered a single computational unit corresponding to a service request from the end - user. The different tasks may have different priorities and constraints, making them dependent upon each other. There may

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.13

be independent tasks in different data flows that can be serially executed. Task scheduling in a distributed computing environment to allocate computing resources in a cloud computing environment to different tasks optimally while meeting the quality of the service to end users as per service level agreement.

Task scheduling finds an optimal pair of fog node and end user task [8]. It aims to select an effective utilization of fog nodes for executing user tasks as per required deadlines and service level agreement. Fog computing contains a fog manager component responsible for making task scheduling decisions efficiently. The placement of fog managers in the fog computing environment is depicted in Figure 2 based upon the most common aspects discussed in the literature.

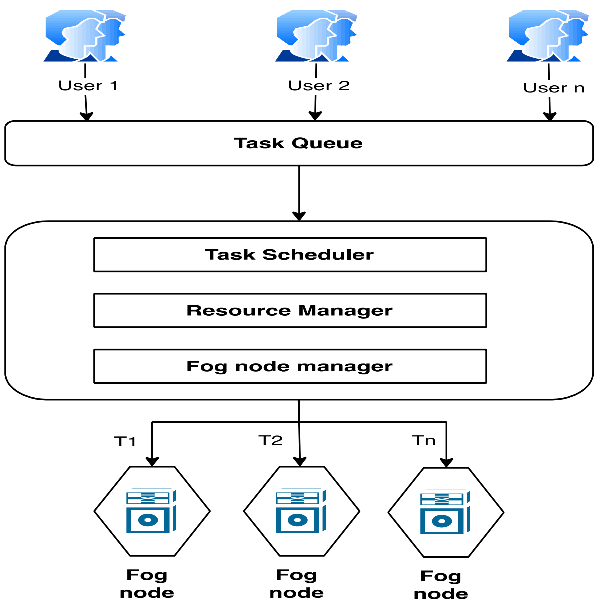

Figure 2. Task scheduling

It can be noted from figure 2 that managers receive end user requests through IoT and sensor devices and assign them different priorities with the help of a request evaluator [9,10]. The prioritized tasks are further passed to fog nodes or cloud nodes for their execution depending upon different factors like communication requirement, computational overhead, computing source requirement [10-13]. A task scheduler handles these tasks by selecting an appropriate combination of task and fog nodes that schedules different tasks on fog nodes for their execution in an effective resource method.

The task scheduler also enables the management of different computing resources for scheduling tasks in a fog computing environment. The task scheduling problem is considered as NP complete problem [14]. NP complete problems have no solution that can be completed in polynomial time.

Several techniques have been proposed for obtaining effective task schedules in a fog computing environment. This work presents a comprehensive review of different task scheduling methods proposed in the fog computing environment in recent years. It investigates different task scheduling methods and compares them in multiple dimensions. It summarises different methods and describes the pros and cons of different test scheduling methods.

The rest of the paper is structured as follows. Section II presents recent task scheduling methods in the fog computing environment. Section III provides classification of task scheduling methods. Section IV investigates recent task scheduling methods and highlights their pros and cons. Section V highlights critical challenges and provides promising research directions. Finally, section VI concludes the paper at the end.

Related Work

This section describes different research articles related to task scheduling methods in the fog computing environment. Many significant surveys have been conducted for investigating different parameters related to task scheduling methods in fog computing. Perera et al. [15] conducted a fog calculation methods survey and investigated their issues. They identified ten primary specifications and essential characteristics of fog computing. They provided a comprehensive comparison of thirty research studies. However, their study was not systematic. They have not clarified the criteria for considering different research studies in their survey.

Sharma and Rani [16] investigated different dimension task scheduling methods in fog computing, including their features, benefits, and limitations and compared them. They concluded that the scheduling algorithm must be located in the operating system for its evaluation. However, they have not conducted their service systematically. They have not been clearly described. Selection of articles, classification of articles and not provided future working their survey paper.

Tsai et al. [17] reviewed various metaheuristic based task scheduling methods for cloud computing. But they ignored the quality of service parameters in their comparative study.

Similarly, Zhan et al. [18] investigated different scheduling algorithms at different layers in the cloud and fog computing environment, such as virtualization, deployment, and application. Their main focus was on evolutionary algorithms. Kalra et al. [19] focused on cloud and grid computing environments and investigated metaheuristic based task scheduling methods. In contrast, Madni et al. [20] analyzed the metaheuristic based method used for allocating computing resources in the IaaS cloud environment. They stated that there are many issues in computing resource allocation methods. They also describe different performance metrics.

Arunarani et al. [21] conducted a survey of task scheduling methods in the cloud computing environment. They classified Different techniques based upon their applications and performance metrics.

Singh et al. [22] performed a comparative analysis of bio inspired and swarm intelligence based metaheuristic methods for different scenarios based upon the type of task, scheduling objective and simulator characteristics.

Kumar et al. [23] reviewed scheduling methods in a cloud computing environment based upon heuristic and metaheuristic algorithms. They described different issues in scheduling algorithms in the cloud computing environment.
Alizadeh et al. [24], Singh et al. [25] Hosseinioun et al. [26] and Sidhu et al. [27] also investigated task scheduling methods in one another way. These review papers compare different studies based upon their focus scheduling methods. They proposed different ways to classify task scheduling methods, such as static and dynamic, heuristic, metaheuristic, and hybrid algorithms. It also described a few open issues and suggested promising research direction in task scheduling in the fog computing environment.
This work aims to present a comprehensive and updated review of task scheduling methods in a fog computing environment to identify significant research gaps with their future scope. The identified gaps can be the promising direction for fellow researchers to develop effective task scheduling methods in fog computing environments.

2. Classification of the Scheduling Strategies

Task scheduling methods in a fog computing environment can be broadly classified into the following categories [28].

- Static task scheduling methods
- Dynamic task scheduling methods
- Hybrid task scheduling methods

Static task scheduling methods require the availability of task requirements to the task scheduler before the desi cutting scheduling policy. The task scheduler computes task requirements before the start of any scheduling process. In this scenario, the tasks are submitted to the system without dependency on the states and accessibility of computing resources.
The most common task scheduling methods in this category are the First Come First serve scheduling method and Round Robin method.
First Come First served scheduling method is a straightforward scheduling method that considers the incoming time of the task without considering any other parameter [29].
The Round Robin method is a variation of the First Come First served scheduling method that allocates a fixed time slot to each task execution. This method in harness is the probability of each task to finish its execution in a given time slot. The task requires execution time more than the specified time slot. It is appended to the task queue for waiting still. All task takes their time slots for execution [30].
Opportunistic load balancing method for task scheduling allocate the closest available machine to execute each task. This method attempts to keep all available machines occupied for executing tasks. The potential benefit of this method is its easy implementation without any additional computational overhead. Other significant static task scheduling methods are maximum completion time and minimum execution time. These methods are mostly suitable for a heterogeneous distributed environment. Minimum execution time-based task scheduling involves executing the task on a machine leading to short test execution time. However, it does not consider computing resource availability e while assigning tasks [31-33].
In contrast, a minimum completion time-based scheduling algorithm assigns task execution to a machine that leads to the shortest completion time. Min min task scheduling method schedule the most minor task to the first available machine resulting in minimum execution time. However, the major limitation of the min min task scheduling method is that it increases the task's execution time. Max min touch scheduling method is similar to min min scheduling method. It overcomes the problem of min min task scheduling method by reducing the execution time of the task. But, it allocates the most significant task to the first available machine for its execution, leading to short test execution time.
Dynamic scheduling methods are developed based upon the arrival of the task at a particular time and the state of the system machine. These methods can consider one task at a time or a group of tasks simultaneously. Accordingly, these methods can be classified as online or Batch Mode. Scheduling algorithm in dynamic scheduling category Min Min, Max Min, Round Robin, suffrage heuristics, minimum execution time, opportunistic load balancing scheduling method, GA, IA, SA, PSO, ACO, MFO, Cuckoo, SSO, Bee life, and Tabu search.
Different algorithms such as minimization maximization methods, multi - objective evolutionary algorithms, and energy aware algorithms have been proposed in the hybrid task scheduling category. Hybrid scheduling methods combine existing algorithms' benefits in developing an optimal solution for task scheduling. For example, Abdullah and Othman, 67 proposed a scheduling approach using min min algorithm that takes task scheduling the seasons on the basis of server load and user task priorities. In their approach, they suggested to divide users into two groups called Normal users and privileged users. They demonstrated that their approach could improve resource usage, user satisfaction, and response time.

TABLE 1 SUMMARIZES THE ABOVE-MENTIONED TASK SCHEDULING METHODS.

Scheduling strategy	Concept applied
Static	First Come First served method
	Max min method
	Min min method
	minimum completion time method
	minimum execution time method
	Opportunistic load balancing method
	Round Robin method
Dynamic	Cross entropy
	GA
	IA
	SA
	PSO
	ACO
	MFO
	Cuckoo
	SSO
	Bee life

3. Task Scheduling Strategies

Many research efforts have been invested in developing effective and efficient task scheduling in fog and cloud computing environments. Different techniques are used in task scheduling structure as a stick method heuristic method, metaheuristic method, hybrid method etc. For example, Wu et al. [34] applied the concept of energy minimization scheduling based upon working energy first and Idle energy first e in developing a task scheduling approach. They focused on minimizing fog nodes' power consumption by finding an optimal solution. They proposed that all tasks can be II scheduled to slow nodes in a fog computing environment that will help to reduce idle energy wastage in fast nodes of fog computing. In order to decrease overall power consumption, the authors used both concepts to find the best possible task schedule in the fog computing environment.

Similarly, Hoang et al. [35] also proposed a heuristic-based approach for task scheduling by dividing the fog layer into two different regions. There exists a fog node called manager to manage each region. The authors target to minimize transmission and computational latency by reducing the completion time of the task. The completion time of the car depends upon computing resources service time task completeness. The authors proposed to sort all tasks in increasing order of their latency.

Several researchers have proposed optimization - based algorithms to optimize task scheduling in a fog computing environment. They used many optimizations and meta heuristic algorithms such as Moth flame optimization algorithm, ant colony optimization algorithm, swarm optimization-based algorithm, evolutionary algorithm artificial immune system based algorithm, bee life algorithm, and many more. For example, Ghobaei et al. [36] suggested task scheduling solution in a fog computing environment using a Moth optimization algorithm. They demonstrated an optimal use of computing resources by scheduling different fog node tasks.

Sujana et al. [37] used a trust - based stochastic scheduling method to ensure efficient task scheduling and provide security. They attempted to find an optimal virtual machine task for scheduling tasks in fog computing. They focused on stochastic level factors for allocating appropriate priorities to the task in a fog computing environment based upon the average computation capacity of virtual machines, inter task communication time, and execution time.

In contrast, Wang et al. [38] addressed the issue of overloading and balancing network traffic based upon immune network theory. They attempted to adjust the overloaded nodes to the neighbouring fog nodes. Their proposed approach comprises two phases dealing with non specific immunization phase and a specific immunization phase. The former phase deals with the overloading issue by determining the fog node's computing resource capacity and the overloading task for execution. The later phase involves a fog scheduler treated as immune cells for formulating scheduling policy after recognizing features of the task. This approach broadcasts the network's overloading information and uses forward and backward propagation methods to optimize scheduling policies with minimum execution time.

Wang and Li [39] proposed a hybrid metaheuristic approach by integrating ant colony optimization and particle swarm optimization algorithms for task scheduling in the fog computing environment. This approach focused on constraints about Terminal devices in the fog network. The main objective considered in this work was the overhead of the terminal devices. They demonstrated that their approach has improved by considering the global state of the network compared to the local state in the network.

Similarly, Rahbari et al. [40] also proposed a hybrid approach targeting network security by considering security objectives, integrity, authentication, and confidentiality security objectives. This approach comprises two phases. These phases are known as training phase and testing phase. The training process allocates workload to computing resources based upon different optimization algorithms, genetic algorithm, ant colony optimization algorithm, particle swarm optimization algorithm, and stimulating annealing. During the test phase, this approach used data mining classification methods for selecting appropriate optimization algorithm obtained during the training phase. The authors mainly focused on each algorithm's power consumption, network bandwidth utilization, and execution cost to find the optimal solution.

Nazir et al. [41] used the cuckoo optimization method for load balancing and achieving energy efficiency by scheduling different tasks in a fog computing environment. They are so applied k - means clustering method for determine meaning overused fog nodes.

Boveiri et al. [42] proposed a meta heuristic Grave scheduling algorithm based upon min max ant system. This approach comprises different phases in effectively providing task scheduling in the fog computing environment. They considered multiple processor environments consisting of heterogeneous fog nodes in their experiments. They demonstrated their approach superiority in comparison to existing scheduling methods.

Sun et al. [43] proposed a Framework for a fog computing environment comprising three layers, terminal layer, at layer and core layer. They proposed computing resource scheduling in two levels. The first level involves computing resource scheduling among fog clusters, and the second level consist of scheduling among fog nodes and respective fog clusters. Since task scheduling is an NP hard problem. The authors used the NSGA II algorithm for addressing multiple and conflicting objectives. In this work, the authors focused on the overall task execution stability and service delay. They experimentally demonstrated that service delay improved using their approach compared to other models. Similarly, overall task execution stability has also improved with their proposed approach.

Similarly, Isard et al. [44] also solved task scheduling problems by considering different regions consisting of heterogeneous cloud and fog nodes. They used a heuristic method for selecting area managers using the decentralized voting method. Therefore, cost on optimizing total cost for executing tasks and power consumption.

Dang and Hoang [45] proposed a task scheduling method based upon an optimization algorithm for minimizing task completion time. They proposed to divide problem space into some areas containing physical places. Each area contains dynamic fog devices. It allows the movement of fog devices from one area to another.

Rasheed et al. [46] analyzed the fog cloud framework related to intelligent grid power distribution. They observed that the integration of the fog cloud platform helps generate transmitting and distributing the power securely and efficiently. In their experiment, they considered six geographical areas. Choudhari et al. [47] proposed an approach scheduling task in fog computing environment considering task priorities. They assumed that the fog layer contains fog nodes considered as small data centres. Fog servers act as managers for managing and allocating computing resources efficiently. Fog nodes can exchange information regarding workload and computing resource allocation. Each client transmits the job to the adjacent Fog server in this approach. The fog server computes the deadline and accordingly queue up the task to execute in the remaining time. Otherwise, the task gets rejected. In this approach, multiple queues are maintained as per the task's priority as a medium, high and low priority tasks. However, this approach suffers from the limitation of considering only deadlines while scheduling the task.

Kabirzadeh et al. [48] proposed a hybrid approach consisting of genetic algorithm, particle swarm optimization and ant colony optimization algorithm for allocating resources among the following notes to enhance computing resource utilization of fog computing environment. The authors divided scheduling objectives into service users and providers and considered multiple parameters such as makespan, deadline, budget, cost, and security. The other parameters considered are power consumption, effective load balancing, and computing resource utilization. The authors compared their hybrid approach with the existing indivisible optimization algorithms, genetic algorithm, ant colony optimization algorithm, stimulating annealing algorithm and particle swarm optimization algorithm in different parameters such as power consumption, network bandwidth and execution time. They demonstrated that the hybrid approach significantly improved power consumption compared to the other heuristic algorithms. They have also validated their approach for reducing execution time and cost in comparison to the other algorithms.

Yin et al. [49] propose an approach based upon containers without considering hypervisors and virtual machines. They reported an improvement in start time and performance using containers in the virtual machine's state. This approach divided the task processing using containers in two different phases. The first phase determines the task to be executed at fog level or cloud level depending upon deadline and execution time. The 2nd phase involves allocating containers to the task executed on fog nodes to maximize computing resource utilization.

Rahbari and Nickray [50] suggested a scheduling approach based upon a knapsack optimization algorithm. They focus on minimizing latency, improving performance, and reducing power consumption. They considered makespan, workload optimization, power consumption and Virtual Machine optimization as objectives of their approach. The simulated results demonstrated that the proposed approach leads to less power and network fan width compared to conventional methods like the ordinary knapsack method and the First Come First serve method.

Liu et al. [51] presented an adaptive double fitness genetic task scheduling method. In this approach, the author proposed to allocate task to fog nodes based upon their computing capability, delay requirements and communication capacity. The authors attempted to optimize IoT devices' communication cost and execution time. Their experimental results demonstrated that the proposed method could reduce communication cost and makespan compared to traditional methods.

Wang et al. [52] suggested a decentralized approach based upon the immune system. They considered the network a framework consisting of computing nodes and can schedule their decisions independently. The decentralized approach can address the issues of computational and communication bottleneck and single point of failure in the centralized system.

Bitam et al. [53] also used the bees life algorithm for scheduling tasks in the fog computing environment. The author considered the live - streaming task at fog nodes residing at network edges. The jobs are further separated into a group of tasks that are scheduled for their execution on fog nodes. The experimental results validated that their approach could reduce execution time and improve the efficiency of the system compared to other conventional methods.

Oueis et al. [54] introduced a task scheduling method that allocates different tasks in a distributed way. A least fog computing resources can be allocated to the short CPU time execution while meeting service level agreement. This method distributed tasks among fog computing nodes for obtaining trade-off among memory allocation and execution time.

Intharawijitr et al. [55] used new communication methods, particularly for 5G cell networks, to reduce computing delay in the cloud computing environment. The authors proposed a fog computing design to obtain improved network performance by optimally scheduling different tasks. They use three types of task scheduling policies in their approach. The first scheduling strategy involves random policy that selects fog network randomly for executing the task on a regular distribution basis. The second strategy is the minimum delay method that fog nodes with the minimum delay time depending upon the system's current status. The third strategy was the maximum amount of remaining computing resources of fog nodes. Experimental results demonstrate that the minimum delay policy can produce the best network performance due to the accessibility of computing resources.

Deng et al. [56] suggested a workflow for task scheduling in a fog cloud computing environment for achieving trade-off between transmission delay and power consumption. They proposed determining the power consumption of fog devices based on workflow assignment and operating frequency. They divided the workflow scheduling problem into sub problems. Each problem is further solved using an optimization method. The first such problem provides a trade-off between communication delay and power consumption using the convex optimization method. The second subproblem applies integer nonlinear programming further to provide trade-off between communication delay and power consumption. Finally, the third subproblem is solved for achieving optimized data transfer.

Pham and Huh [57] proposed three-layer architecture fog computing environment. The proposed architecture obtained a trade-off between financial cost and execution time of the tasks. This e architecture contains three layers. The lowest layer comprises IoT devices that forward user requests to the higher layers. The fog layer accepts the requests and performs them based on computing resources and transferring the rest to the cloud servers. They proposed a distributed system for distributing workflow in the fog cloud environment.

Li et al. [58] proposed an approach using fuzzy clustering methods and particle swarm optimization algorithms that schedule fog computing resources based on computing standardized and normalized features of computing resources. This approach can reduce the search space of computing resources using a fuzzy c means clustering algorithm and particle swarm optimization algorithm. It works in two phases. The first phase initializes the particle population. Each particle indicates a set of clusters centre generated arbitrarily. It is followed by a competition of membership Matrix and respective fitness values. The fuzzy c - means clustering method gets trapped into local minima. This approach uses a particle swarm optimization algorithm with the benefit of fast convergence and global optimization. The experimental results indicate that this method leads to fast convergence and higher accuracy in comparison to the individual fuzzy clustering method and particle swarm optimization method.

Nguyen et al. [59] investigated a bag of task approach for optimizing task scheduling problems in a fog cloud computing environment. They focused on optimizing operating costs and execution time using genetic algorithms. They attempted to trade-off between the cost for completing tasks and execution time in fog cloud computing.

Wang and Li [60] introduced a task scheduling policy called the hybrid heuristic method to address restricted computing resources and extraordinary power consumption issues at terminal devices in the fog computing environment. The proposed method involves using an improved ant colony optimization method and improved particle swarm optimization method to solve task scheduling problems and minimize power consumption and delay. They validated their approach in terms of reliability, power consumption and completion time. Results indicate that their approach is better than individual approaches by achieving improved completion time, power consumption and better reliability.

Benblidia et al. [61] proposed a fuzzy quantified ranking based method for task scheduling in the fog computing environment. This method performed task scheduling activity based on fog nodes' ranking based on their needs and task preferences. This approach has effectively optimized power consumption, execution delay, and user satisfaction.

Jamil et al. [62] proposed a scheduling approach to support service provisioning in the Internet of everything for the fog computing environment. They analyzed optimal scheduling of the request fog devices based upon their computing capability. The experimental results validated that their approach has reduced delay and network utilization to a significant level.

Zhao et al. [63] proposed a multi layered architecture for dynamic modelling of content delivery wireless networks. This network contains heterogeneous devices that differ in their processing capabilities, storage capabilities and network communication. The proposed architecture improved network throughput, fairness in response and service delay. This approach has low computational overhead and provides a trade-off between service delay and network throughput.

Yang et al. [64] introduced an approach for managing communication and computing resources in the fog computing environment. They focused on optimizing computational offloading in fog computing. Firstly, they specified a random system for finding the location of the fog node, the link between them, their traffic needs. They assigned a local fog controller that dynamically schedules the task in an online way. The local fog controller schedules the task ok after each fixed interval based upon the incoming task and status of the network. In determining the dynamic values of CPU cycle frequency, power consumption for task scheduling operation. They experimentally demonstrated that their approach outperformed conventional algorithms regarding network performance, delay and power consumption under different scenarios.

Wan et al. [65] introduced an energy aware load balancing and scheduling method for a fog computing environment. They applied particle swarm optimization method to obtain an optimal task scheduling solution in NP hard problem. They applied this approach for candy packing lines to generate a scheduling table and optimal workflow.

Jie et al. [66] introduced an online approach for task scheduling based upon the Repeated Stackelberg Game method. In their approach, edge service providers are considered long - term followers and users in each iteration as short - term followers. This approach contains three layer architecture: user, edge service provider, and cloud service provider. The service provider layer contains many small data centres that are graphically distributed. The service provider layer makes the protections for fog computing resource requirements based upon their historical requirements in previous iterations. They divided each task into multiple tasks and mapped them to computing resources available in edge data centres. If there are no suitable service completing resources, it is forwarded to the cloud service provider layer. Experimental results validated their approach in computing resource utilization and execution time compared to other conventional algorithms.

Cardellini et al. [67] introduced a quality of service aware scheduling method for processing data streams. This method contains a worker monitoring component, quality of service monitoring component and an adaptive scheduling component. The worker monitoring component calculates the input and output rate for each worker. It behaves as a counting machine for executing multiple tasks on fog nodes. A local database is used to store input and output rates by using adaptive scheduling components. The quality of service monitoring component approximates quality parameters like delay. It computer internal performance and intra node performance related information. The collected information is transmitted to the distributed adaptive scheduling component to implement the scheduling strategy. If the task can be executed efficiently, then task is assigned to it. Experimental results validate this approach over other methods.

Ningning et al. [68] used the graph partitioning method for developing a load balancing approach. This approach can allocate computing resources to multiple fog nodes based upon task requirements. In this approach, physical nodes are divided into virtual machines groups, and virtual machine nodes offer services for end users through graph partitioning. This approach generates a minimum spanning tree from a graph with those edges removed that cannot provide sufficient computing resources. The authors mainly focused on execution time.

Zeng et al. [69] proposed a software - defined system for keeping task images in the storage server. They mainly focused on scheduling activity for minimizing completion time of the task. Here they proposed that computing will be performed on embedded devices.

Sharma and Saini [70] presented a four - tier framework supporting the scheduling and workload balancing in the fog computing environment. The first tier contains IoT devices. The second tier has different applications classified as low priority and high priority based upon the dual Fuzzy Logic method. The fuzzy logic method considered input like task size, arrival time, minimum execution time, and maximum completion time. The highly prioritized tasks are sent to tier 3, containing a new fog arrangement. In this approach, fog nodes are clustered using the K means clustering algorithm. They evaluate their approach using a real - time application based on schedule, response time, power consumption and workload balancing ratio.

Gazori et al. [71] introduced a task scheduling approach for minimizing long - term service delay and computational cost. In this approach, the authors suggested using the reinforcement learning method and presented a double deep Q learning based task scheduling method. They conducted experiments by considering propagation, waiting, transmission and execution delay of different tasks and why allocation of user tasks to virtual machines. The experimental results validated their approach over existing algorithms.

Abdelmoneem et al. [72] presented an IoT design for healthcare application for mobility aware scheduling algorithm and computing resource allocation protocols. The authors mainly focused on minimizing schedule time based upon significant characteristics such as critical level and response time. This approach supports patients' mobility using adaptive received signal strength-based handoff method. This approach has been validated for balancing the distribution of cast execution dynamically based upon movement of the patient and temporal spatial residual data. Table 2 summarizes the above cited studies.

4. Challenges and Future Directions

Several task scheduling methods have been described above in the fog computing environment. Different researchers focus on different parameters for improving service quality, network performance, power consumption, etc. Some of them also emphasized makespan, workload balancing, financial cost, response time, computing resource utilization and effective utilization of energy In fog and cloud computing. Some researchers

validated their approaches based upon simulation; however, others did not evaluate experimentally [80-81].
To overcome this limitation, appropriate attention must be given to the following issues in future, particularly for the fog computing environment.

1) Workload characterization has a significant role in developing efficient scheduling, energy saving and resource provisioning strategies [82]. Therefore, understanding the workload received from IoT devices in a fog computing environment can be an active research area for developing effective scheduling, energy saving and resource provisioning methods.
2) Mobile fog computing has the benefit of offering the quality of service and power consumption [83-85]. However, little attention has been given to this area of mobile fog computing and its mobility aspects.
3) Scalability is a challenging task in the fog computing environment. The fellow researchers can develop scalable algorithms with respect to the increasing number of IoT devices and networks [84-86].
4) Data stream-based applications have emerged in many real life domains. However, fog computing can be a promising direction for addressing real time data stream processing challenges [87].
5) Limited infrastructure exists for validating a real fog computing environment. Therefore, many fog computing research has been validated using simulation tools. Therefore, there is a need to develop a large - scale and real - time testbed for validating fog computing approaches [84-86].
6) Fog devices have restricted processing and storage abilities and are heterogeneous in nature [88]. These Limited parameters make the task scheduling in a fog computing environment more challenging. Therefore, intelligent task scheduling methods must be e developed for improving network performance in the fog computing environment.
7) Fog computing environments have pin developed to support real time latency sensitive applications [89]. Therefore, reliability must be maintained. So, reliability can be a promising research direction in this area.
8) Most IoT devices and fog devices are limited in their battery storage [86]. So effective power management solution can be a future research direction.
9) Security is considered an essential aspect of fog computing in management that requires secure communication between IoT devices and data centres with limited computing resources in the fog layer [86,87]. Most fog computing devices can suffer from security problems because of insecure deployment in the restricted environment. Any IoT device can pretend for legitimacy. Therefore, the security aspect must be considered seriously. To address the issue of security privacy protection, a template can be used in scheduling processes before granting resources to different tasks.
10) Quality of service and service level agreement is another issue in fog computing environment related to satisfaction and non satisfaction of end users and service providers [5]. Researchers investigated the quality of service in three aspects related to performance, reliability, and cost.
11) However, most researchers ignored the quality of service and service level agreements in their research fog computing environment.
12) Fog computing contains many heterogeneous and diverse fog devices that lead to heterogeneity in collecting data, the format of the data and processing of the data [89]. Fog devices such as switches, routers, and gateways have different storage and processing capabilities. Therefore, fog computing policies such as task scheduling and resource provisioning resource allocator must be e developed considering the heterogeneity of fog computing devices.
13) Fog scheduler involves assignment and updating task priorities as per their dynamic needs [88]. However, assigning priorities is challenging based on different parameters like that line, maximum completion time, minimum execution time, etc.

TABLE 2. SUMMARY OF TASK SCHEDULING METHODS IN FOG COMPUTIN

Study	Approach	Evaluation criteria	Pros	Cons
Abdelmoneem et al.[72]	Task scheduling considering mobility	• Resource utilization • Response time	• Improves power consumption • Improves response time • Improves scheduling time	• Not investigated deadlines • Not investigated resource issues
Al Ahmad et al.[79]	energy aware fog server selection method	• Resource utilization • Power consumption	• Improves scalability • Improves power consumption • Improves mobility	• Not investigated user satisfaction • Not investigated Availability
Benblidia et al.[61]	User preference and fog nodes features aggregation	• Execution time • Power consumption • User satisfaction	• Decreases power consumption • Improves execution time	• Cost model not described
Bitam et al.[53]	Task scheduling based on bees Life method	• SLA • Resource utilization • Execution time	• Improves efficiency • Improves performance • Minimum execution time	• Not investigated communication overhead • Static in nature
Bittencourt et al.[74]	Mobility aware application scheduling	• Cost • Response time	• Satisfies mobile user needs	• Application prioritization not described

Cardellini et al.[67]	Distributed QoS aware data stream scheduling	• Delay • Availability	• QoS aware scheduling	• High complexity • Resulted operation instabilities
Choudhari et al.[47]	Task scheduling	• Cost • Execution time • Response time	• Combines the existing scheduling algorithms • Decreases response time • Improves cost	• Only considered priority based on execution deadline
Dang and Hoang[45]	Task scheduling	• Deadline time • Delay • Execution time • Resource utilization • Through put	• Determines the efficient method to select the manager fog node in regions • Improves the resource utilization rate • Reduces of tasks' completion time • Simple computations	• Not investigated factors such as power and task execution cost
Deng et al.[56]	Resource allocation for workflow investigated cloud servers	• Network bandwidth • Delay	• Presents a two - level method for allocating resources • Improves the amount of information exchange requests	• High computational complexity
Gad - Elrab and Noaman[76]	Fog node selection optimization	• Cost • Deadline • Power consumption	• Improves make span • Improves cost	• Not investigated communication costs • High computational complexity
Gazori et al. [71]	Task scheduling	• Cost • Deadline • Delay • Power consumption • Response time	• Improves power consumption • Improves delay	• Not investigated user satisfaction • Not investigated security
Intharawijitr et al.[55]	Communication methods based on scheduling policies	• Resource utilization • Delay	• Reduces computation • Improves communication delay	• Considered 5G only
Jamil et al. [62]	Job scheduling	• Delay • Network bandwidth • Power consumption	• Improves waiting time • Improves network usage • Improves delay	• Not investigated computing resource constraints • Not investigated task priority
Jie et al.[66]	Task scheduling using Repeated Stackelberg Game approach	• User satisfaction • Resource utilization • Execution time • Delay • Cost	• Improves the execution time • Improves efficiency	• Not investigated deadlines
Kabirzadeh et al.[48]	Resource scheduling and allocation	• Cost • Deadline • Delay • Power consumption • Resource utilization • Security	• Decreases power consumption • Improves execution time • Minimum cost	• No change in power consumption in comparison to GA method
Li et al.[58]	Resource clustering and scheduling	• User satisfaction • Network bandwidth	• Quick convergence • Improves accuracy	• Not investigated real time resource changes
Liu et al.[51]	Dynamic task scheduling	• Network bandwidth • Execution time • Delay • Cost	• Decreases communication cost • Improves make span	• No investigation of task prioritization
Nguyen et al.[59]	Task scheduling based on evolutionary algorithms	• User satisfaction • Execution time • Cost	• time optimization • Trade-off between make span and cost	• Ignored power consumption of transmission • Not investigated cost • Not investigated the computing resource limitations
Ningning et al.[68]	Resource allocating Load balancing approach using graph partitioning	• Execution time	• Applies graph theory	• Performs Sub optimally for dynamic work load balance
Oueis et al.[54]	Request execution localization	• User satisfaction • Delay • Deadline	• Small delay • High power efficiency •	• Performance degrades for high - density computing
Pham and Huh[57]	Dynamic task scheduling based on heuristic and graph theory	• Network bandwidth • Execution time • Cost	• Improves communication overhead • Improves costs • Maximizes resource usage	• Not investigated delay sensitive application needs • No supports for fog / cloud services
Rahbari and Nickray[50]	Task scheduling using Knapsack game based optimization methods	• Cost • Delay • Power consumption • Reliability • Security	• Decreases power consumption • Improves delay • Improves network performance	• Ignored costs • Not investigated communication factors • Not investigated memory issues
Rahbari et al.[73]	Resource scheduling investigated security	• Cost • Deadline • Delay • Power consumption • Security • Resource utilization	• Improves execution time • Decreases power consumption	• Takes longer than the ACO algorithm • Enhanced operational overhead
Rasheed et al.[46]	Electricity distribution smart grid investigation	• Cost • Response time	• Decreases response time • Improves cost	• Not investigated memory issues • Static in nature
Sharma and Saini[70]	Load balancing Delay aware task scheduling	• Deadline • Execution time • Power consumption • Response time	• Improves power consumption • Improves response time • Improves scheduling time	• Considered priority based on deadline only

Sun et al.[43]	Resource scheduling	• Delay • Execution time • Reliability • Resource utilization	• Improves service delay • Improves stability of task execution	• Not investigated resource allocation cost • Unable to find first level scheduling method
Sun et al.[78]	Job scheduling approach	• Response time • Resource utilization • Execution time	• Resource usage • Improves execution time	• Not investigated network bandwidth
Verma et al.[75]	Managing workflow amount fog and cloud Task scheduling	• Availability • Deadline • Delay • Network bandwidth	• Improves bandwidth • Improves deadline • Increases resource availability • Optimizes delay	• Communication factors not described • Cost factors not described • Not investigated QoS factors
Wan et al.[65]	Load balancing Power aware task scheduling	• Power consumption • Delay	• Improves device performance • Improves transmission delay • Optimizes computational costs • Optimizes power consumption	• workload of devices • Investigated only power consumption
Wang and Li[60]	Task scheduling based on IPSO and IACO	• Reliability • Power consumption • Delay	• Improves reliability • Improves completion time • Decreases power consumption	• Increases complexity with more user requests
Wang et al.[52]	Task scheduling based on human immune system	• Reliability	• Prevents generating communication and computing blockages and single point of failure issues • Improves the tasks' finish time • Decreases algorithm iterations	• No investigation of execution of tasks and priortization • Cost model not described
Yang et al.[64]	Computation offloading Dynamic computing resource management	• Power consumption • Delay	• Improves power consumption • Improves delay jitter • Improves delay in performing operations	• Considered homogenous environment only
Yin et al.[49]	Focused containers for task scheduling	• Resource utilization • Execution time • Delay • Deadline	• Applies containers • Decreases the tasks' execution times • Increases the accepted tasks	• Preparation and application of containers are described
Ying Wah et al.[77]	Task scheduling	• Network bandwidth • Deadline • Cost	• Reduces cost • Investigates deadlines in the execution of tasks	• Not investigated power consumption • Not investigated resource issues
Zeng et al.[69]	Task scheduling Task positioning	• SLA • Reliability • Delay • Cost	• Considered transmission delay between different layers	• Not investigated memory usage • Not investigated deadlines • Increased computational complexity
Zhao et al.[63]	Models content delivery wireless network	• Delay • Network bandwidth • Throughput • User satisfaction	• Bandwidth allocation dynamically • Improves system performance	• Not considered traffic distribution type • Not investigated the network dynamicity • Not observed request contents

5. Conclusion

This paper presents a comprehensive review of task scheduling methods, investigates different methods, and highlights their advantages and disadvantages, particularly for the fog computing environment. The primary research studies of task scheduling in fog computing environments have been compared in multiple dimensions, including evaluation criteria, advantages, and disadvantages.

It can be concluded from the studies mentioned above that there exist many search issues that must be addressed adequately in the fog computing environment. To that end, this paper provides a comprehensive list of future research work in the fog computing environment related to different aspects such as heterogeneity, diversity, security, power consumption, makespan, load balancing, financial cost, response time, execution time at completion time that have been ignored by most researchers in the field. Therefore, by integrating different approaches and considering significant performance factors, it is possible to enhance the effectiveness of scheduling algorithms in the fog computing environment.

References

[1] Buyya, R., James, B. and Goscinski, A.M. (2010), Cloud Computing : Principles and Paradigms, Vol. 87, John Wiley and Sons.

[2] Dikaiakos, M.D., Katsaros, D., Mehra, P., Pallis, G. and Vakali, A. (2009), “Cloud computing : Distributed internet computing for IT and scientific research”, IEEE Internet Computing, Vol. 13 No. 5, pp.10 - 13.

[3] de Carvalho, E.R., da Mota, A.E.A.S., de Souza Martins, G.M., Bastos, L. D S. L. and Melo, A.C.S. (2017), “The current context of lean and six sigma logistics applications in literature : a systematic review”, Brazilian Journal of Operations and Production Management,Vol. 14 No.4, pp.586 - 602.

[4] Yang, Xin, and Nazanin Rahmani. "Task scheduling mechanisms in fog computing : review, trends, and perspectives." Kybernetes (2020).

[5] Singh, R. M., Awasthi, L. K., & Sikka, G. (2022). Towards Metaheuristic Scheduling Techniques in Cloud and Fog : An Extensive Taxonomic Review. ACM Computing Surveys (CSUR), 55 (3), 1 - 43.

[6] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya. 2016. Fog computing : Principles, Architectures, and Applications. Internet of Things. Morgan Kaufmann (2016), 61–75.

[7] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue. 2019. All one needs to know about fog computing and related edge computing paradigms : A complete survey. Journal of Systems Architecture 98, (2019) 289–330.

[8] Kaur, N., Kumar, A., & Kumar, R. (2021). A systematic review on task scheduling in Fog computing : Taxonomy, tools, challenges, and future directions. Concurrency and Computation : Practice and Experience, 33 (21), e6432.

[9] Sharma S, Saini H. A novel four - tier architecture for delay aware scheduling and load balancing in fog environment. Sustain Comput Inform Syst. 2019 ; 24 : 100355.

[10] Luo, Juan, Luxiu Yin, Jinyu Hu, Chun Wang, Xuan Liu, Xin Fan, and Haibo Luo. "Container - based fog computing architecture and energy - balancing scheduling algorithm for energy IoT." Future Generation Computer Systems 97 (2019) : 50 - 60.

[11] Naha, Ranesh Kumar, Saurabh Garg, Andrew Chan, and Sudheer Kumar Battula. "Deadline - based dynamic resource allocation and provisioning algorithms in fog - cloud environment." Future Generation Computer Systems 104 (2020) : 131 - 141.

[12] Singh, A., Auluck, N., Rana, O., Jones, A., & Nepal, S. (2019). Scheduling Real - Time Security Aware Tasks in Fog Networks. IEEE Transactions on Services Computing, 14 (6), 1981 - 1994.

[13] Stavrinides, G. L., & Karatza, H. D. (2019). A hybrid approach to scheduling real - time IoT workflows in fog and cloud environments. Multimedia Tools and Applications, 78 (17), 24639 - 24655.

[14] Johnson, D. S., & Garey, M. R. (1979). A guide to the theory of NP completeness. computers and intractability.

[15] Perera, C., Qin, Y., Estrella, J.C., Reiff - Marganiec, S. and Vasilakos, A.V. (2017), “Fog computing for sustainable smart cities : a survey”, ACM Computing Surveys, Vol. 50 No. 3, pp. 1 - 43.arXiv preprint arXiv : 1703.07079.

[16] Sharma, R. and Rani, S. (2019), “Resource scheduling in fog computing : a review”, International Journal of Advanced Studies of Scientific Research, Vol. 4 No. 3.

[17] Chun - Wei Tsai and Joel J.P.C. Rodrigues. 2013. Metaheuristic scheduling for cloud : A survey. IEEE Systems Journal 8, 1 (2013), 279–291.

[18] Z. H. Zhan, X. F. Liu, Y. J. Gong, J. Zhang, H. S. H. Chung, and Y. Li. 2015. Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Computing Surveys 47, 4 (2015), 1–33.

[19] Mala Kalra and Sarbjeet Singh. 2015. A review of metaheuristic scheduling techniques in cloud computing. Egyptian Informatics Journal 16, 3 (2015), 275–295.

[20] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid. 2016. An appraisal of meta - heuristic resource allocation techniques for IaaS cloud. Indian Journal of Science and Technology 9, 4 (2016), 1–14.

[21] A. R. Arunarani, D. Manjula, and V. Sugumaran. 2019. Task scheduling techniques in cloud computing : A literature survey. Future Generation Computer Systems 91 (2019), 407–415.

[22] P. Singh, M. Dutta, and N. Aggarwal. 2017. A review of task scheduling based on meta - heuristics approach in cloud computing. Knowledge and Information Systems 52, 1 (2017), 1–51.

[23] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh. 2019. A comprehensive survey for scheduling techniques in cloud computing. Journal of Network and Computer Applications 143 (2019), 1–33.

[24] Alizadeh MR, Khajehvand V, Rahmani AM, Akbari E. Task scheduling approaches in fog computing : a systematic review. Int J Commun Syst. 2020 ; 33 (16) : e4583.

[25] Singh RM, Awasthi LK, Sikka G. Techniques for task scheduling in cloud and fog environment : a survey. Paper presented at : Proceedings of the International Conference on Futuristic Trends in Networks and Computing Technologies ; 2019 : 673 - 685.

[26] Hosseinioun, P., Kheirabadi, M., Kamel Tabbakh, S. R., & Ghaemi, R. (2020). aTask scheduling approaches in fog computing : a survey. Transactions on Emerging Telecommunications Technologies, e3792. https : //doi.org/10.1002/ett.3792.

[27] Sindhu, V., & Prakash, M. (2019, November). A Survey on Task Scheduling and Resource Allocation Methods in Fog Based IoT Applications. In International Conference on Communication and Intelligent Systems (pp. 89 - 97). Springer, Singapore.

[28] Alizadeh, M. R., Khajehvand, V., Rahmani, A. M., & Akbari, E. (2020). Task scheduling approaches in fog computing : A systematic review. International Journal of Communication Systems, 33 (16), e4583.

[29] Rajesh ME, Mahalakshmi MJ. Optimization of resource allocation using FCFS scheduling in cloud computing. Optimization. 2015 ; 5 (2) : 20 - 26.

[30] El Amrani C, Gibet Tani H. Smarter round robin scheduling algorithm for cloud computing and big data. J Data Mining Digital Humanit. 2018.

[31] Wu X, Huang D, Sun YE, Bu X, Xin Y, Huang H. An efficient allocation mechanism for crowdsourcing tasks with minimum execution time. In : International Conference on Intelligent Computing. Cham : Springer ; 2017, August : 156 - 167.

[32] Li Y, Niu J, Zhang J, Atiquzzaman M, Long X. Real - time scheduling for periodic tasks in homogeneous multi - core system with minimum execution time. In : International Conference on Collaborative Computing : Networking, Applications and Worksharing. Cham : Springer ; 2016 : 175 - 187.

[33] Srinath HMDM. Memory constrained load shared minimum execution time grid task scheduling algorithm in a heterogeneous environment. Indian J Sci Technol. 2015 ; 8 (15) : 15.

[34] Wu HY, Lee CR. Energy efficient scheduling for heterogeneous fog computing architectures. Paper presented at : Proceedings of the IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC),Tokyo, Japan ; Vol.1, 2018 : 555 - 560.

[35] Hoang, D., & Dang, T. D. (2017, August). FBRC : Optimization of task scheduling in fog - based region and cloud. In 2017 IEEE Trustcom/BigDataSE/ICESS (pp. 1109 - 1114). IEEE. https : //doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.36.

[36] Ghobaei-Arani, Mostafa, Alireza Souri, Fatemeh Safara, and Monire Norouzi. "An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing." Transactions on Emerging Telecommunications Technologies 31, no. 2 (2020) : e3770.https : //doi.org/10.1002/ett.3770.

[37] Sujana J, Angela J, Geethanjali M, Raj RV, Revathi T. Trust Model Based Scheduling of Stochastic Workflows in Cloud and Fog Computing. New York, NY : Springer ; 2019 : 29 - 54.

[38] Wang Y, Guo C, Yu J. Immune scheduling network based method for task scheduling in decentralized fog computing. Wirel Commun Mob Comput. 2018 ; 2018 : 1–9.https : //doi.org/10.1155/2018/2734219.

[39] Wang J, Li D. Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing. Sensors. 2019 ; 19 (5) : 1023.

[40] Rahbari, Dadmehr, Sabihe Kabirzadeh, and Mohsen Nickray. "A security aware scheduling in fog computing by hyper heuristic algorithm." In 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), pp. 87 - 92. Ieee, 2017.

[41] Nazir, Saqib, Sundas Shafiq, Zafar Iqbal, Muhammad Zeeshan, Subhan Tariq, and Nadeem Javaid. "Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid." In International Conference on Intelligent Networking and Collaborative Systems, pp. 34 - 46. Springer, Cham, 2018.

[42] Boveiri HR, Khayami R, Elhoseny M, Gunasekaran M. An efficient swarm - intelligence approach for task scheduling in cloud - based internet of things applications.J Ambient Intell Humaniz Comput.2019 ; 10 (9) : 3469 - 3479.

[43] Sun Y, Lin F, Xu H. Multi - objective optimization of resource scheduling in fog computing using an improved NSGA - II. Wireless Personal Comm. 2018 ; 102 (2) : 1369 - 1385.

[44] Isard M, Prabhakaran V, Currey J, Wieder U, Talwar K, Goldberg A. Quincy : fair scheduling for distributed computing clusters. In : Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles. ACM ; 2009, October : 261 - 276.

[45] Hoang, Doan, and Thanh Dat Dang. "FBRC : Optimization of task scheduling in fog - based region and cloud." In 2017 IEEE Trustcom / Big Data SE / ICESS, pp. 1109 - 1114. IEEE, 2017.

[46] Rasheed S, Javaid N, Rehman S, Hassan K, Zafar F, Naeem M. A cloud - fog based smart grid model using max–min scheduling algorithm for efficient resource allocation. In : International Conference on Network - Based Information Systems. Cham : Springer ; 2018 : 273 - 285.

[47] Choudhari T, Moh M, Moh TS. Prioritized task scheduling in fog computing. In : Proceedings of the ACMSE 2018 Conference. ACM ; 2018 : 22.

[48] Kabirzadeh S, Rahbari D, Nickray M. A hyper heuristic algorithm for scheduling of fog networks. In : 2017 21st Conference of Open Innovations Association (FRUCT). IEEE ; 2017 : 148 - 155.

[49] Yin L, Luo J, Luo H. Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. IEEE Trans Industrial Informatics. 2018 ; 14 (10) : 4712 - 4721.

[50] Rahbari, D., & Nickray, M. (2017, November). Scheduling of fog networks with optimized knapsack by symbiotic organisms search. In 2017 21st Conference of Open Innovations Association (FRUCT) (pp. 278 - 283). IEEE.

[51] Liu Q, Wei Y, Leng S, Chen Y. Task scheduling in fog enabled Internet of Things for smart cities. In : 2017 IEEE 17th International Conference on Communication Technology (ICCT). IEEE ; 2017, October : 975 - 980.

[52] Wang Y, Guo C, Yu J. Immune scheduling network based method for task scheduling in decentralized fog computing. Wireless Communications and Mobile Computing. 2018 ; 2018.

[53] Bitam S, Zeadally S, Mellouk A. Fog computing job scheduling optimization based on bees swarm. Enterprise Info Syst. 2018 ; 12 (4) : 373 - 397.

[54] Oueis J, Strinati EC, Barbarossa S. The fog balancing : load distribution for small cell cloud computing. In : 2015 IEEE 81st Vehicular Technology Conference (VTC Spring). IEEE ; 2015 : 1 - 6.

[55] Intharawijitr K, Iida K, Koga H. Analysis of fog model considering computing and communication latency in 5G cellular networks. In : 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). IEEE ; 2016 : 1 - 4.

[56] Deng R, Lu R, Lai C, Luan TH, Liang H. Optimal workload allocation in fog–cloud computing toward balanced delay and power consumption. IEEE Internet Things J. 2016 ; 3 (6) : 1171 - 1181.

[57] Pham XQ, Huh EN. Towards task scheduling in a cloud - fog computing system. In : 2016 18th Asia - Pacific Network Operations and Management Symposium (APNOMS). IEEE ; 2016 : 1 - 4.

[58] Li G, Liu Y, Wu J, Lin D, Zhao S. Methods of resource scheduling based on optimized fuzzy clustering in fog computing. Sensors. 2019 ; 19 (9) : 21 - 22.

[59] Nguyen BM, Thi Thanh Binh H, Do Son B. Evolutionary algorithms to optimize task scheduling problem for the IoT based Bag - of - Tasks application in cloud - fog computing environment. Applied Sci. 2019 ; 9 (9) : 17 - 30.

[60] Wang J, Li D. Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing. Sensors. 2019 ; 19 (5) : 10 - 23.

[61] Benblidia MA, Brik B, Merghem - Boulahia L, Esseghir M. Ranking fog nodes for tasks scheduling in fog–cloud environments : a fuzzy logic approach. In : 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE ; 2019, June : 1451 - 1457.

[62] Jamil B, Shojafar M, Ahmed I, Ullah A, Munir K, Ijaz H. A job scheduling algorithm for delay and performance optimization in fog computing. Concurr Comp : Practice Exp. 2020 ; 32 (7) : e5581.

[63] Zhao S, Yang Y, Shao Z, Yang X, Qian H, Wang CX. FEMOS : fog - enabled multitier operations scheduling in dynamic wireless networks. IEEE Internet Things J. 2018 ; 5 (2) : 1169 - 1183.

[64] Yang Y, Zhao S, Zhang W, Chen Y, Luo X, Wang J. DEBTS : delay energy balanced task scheduling in homogeneous fog networks. IEEE Internet Things J. 2018 ; 5 (3) : 2094 - 2106.

[65] Wan J, Chen B, Wang S, Xia M, Li D, Liu C. Fog computing for energy - aware load balancing and scheduling in smart factory. IEEE Trans Industrial Informatics. 2018 ; 14 (10) : 4548 - 4556.

[66] Jie Y, Tang X, Choo KKR, Su S, Li M, Guo C. Online task scheduling for edge computing based on repeated Stackelberg game. J Parallel Distrib Comp. 2018 ; 122 : 159 - 172.

[67] Cardellini V, Grassi V, Presti FL, Nardelli M. On QoS - aware scheduling of data stream applications over fog computing infrastructures. In : 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE ; 2015, July : 271 - 276.

[68] Ningning S, Chao G, Xingshuo A, Qiang Z. Fog computing dynamic load balancing mechanism based on graph repartitioning. China Comm. 2016 ; 13 (3) : 156 - 164.

[69] Zeng D, Gu L, Guo S, Cheng Z, Yu S. Joint optimization of task scheduling and image placement in fog computing supported software defined embedded system. IEEE Trans Comput. 2016 ; 65 (12) : 3702 - 3712.

[70] Sharma S, Saini H. A novel four - tier architecture for delay aware scheduling and load balancing in fog environment. Sustain Comp : Info Syst. 2019 ; 24 : 100355.

[71] Gazori P, Rahbari D, Nickray M. Saving time and cost on the scheduling of fog - based IoT applications using deep reinforcement learning approach. Future Gen Comp Syst. 2019.1098 - 1115.

[72] Abdelmoneem RM, Benslimane A, Shaaban E. Mobility - aware task scheduling in cloud - fog IoT - based healthcare architectures. Comp Networks. 2020 ; 107348 - 107354.

[73] Rahbari D, Kabirzadeh S, Nickray M. A security aware scheduling in fog computing by hyper heuristic algorithm. In : 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS). IEEE ; 2017 : 87 - 92.

[74] Bittencourt LF, Diaz - Montes J, Buyya R, Rana OF, Parashar M. Mobility - aware application scheduling in fog computing. IEEE Cloud Comp. 2017 ; 4 (2) : 26 - 35.

[75] Verma M, Bhardwaj N, Yadav AK. Real time efficient scheduling algorithm for load balancing in fog computing environment. Int J Inf Technol Comput Sci. 2016 ; 8 (4) : 1 - 10.

[76] Gad - Elrab AA, Noaman AY. A two - tier bipartite graph task allocation approach based on fuzzy clustering in cloud - fog environment. Future Gen Comp Syst. 2020 ; 103 : 79 - 90.

[77] Ying Wah T, Gopal Raj R, Lakhan A. A novel cost - efficient framework for critical heartbeat task scheduling using the Internet of medical things in a fog cloud system. Sensors. 2020 ; 20 (2) : 441.

[78] Sun Z, Li C, Wei L, Li Z, Min Z, Zhao G. Intelligent sensor - cloud in fog computer : a novel hierarchical data job scheduling strategy. Sensors. 2019 ; 19 (23) : 50 - 83.

[79] Al Ahmad M, Patra SS, Barik RK. Energy - efficient resource scheduling in fog computing using SDN framework. In : Progress in Computing, Analytics and Networking. Singapore : Springer ; 2020 : 567 - 578.

[80] Azizi, S., Shojafar, M., Abawajy, J., & Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. Journal of Network and Computer Applications, 103333.

[81] JAMIL, B., IJAZ, H., SHOJAFAR, M., MUNIR, K., & BUYYA, R. (2022). Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions.

[82] Shakarami, A., Shakarami, H., Ghobaei-Arani, M., Nikougoftar, E., & Faraji-Mehmandar, M. (2022). Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review. Journal of Systems Architecture, 122, 102362.

[83] AL-Amodi, S., Patra, S. S., Bhattacharya, S., Mohanty, J. R., Kumar, V., & Barik, R. K. (2022). Meta-heuristic Algorithm for Energy-Efficient Task Scheduling in Fog Computing. In Recent Trends in Electronics and Communication (pp. 915-925). Springer, Singapore.

[84] Singh, R. M., Awasthi, L. K., & Sikka, G. (2022). Towards Metaheuristic Scheduling Techniques in Cloud and Fog: An Extensive Taxonomic Review. ACM Computing Surveys (CSUR), 55(3), 1-43.

[85] Hosseini, E., Nickray, M., & Ghanbari, S. (2022). Optimized task scheduling for cost-latency trade-off in mobile fog computing using fuzzy analytical hierarchy process. Computer Networks, 108752.

[86] Ometov, A., Molua, O. L., Komarov, M., & Nurmi, J. (2022). A Survey of Security in Cloud, Edge, and Fog Computing. Sensors, 22(3), 927.

[87] Hossam, H. S., Abdel-Galil, H., & Belal, M. A. A Survey of Fog Computing: Architecture and Research Challenges.

[88] Chandak, A. V., Ray, N. K., Barik, R. K., & Kumar, V. (2022). Performance Analysis of Task Scheduling Heuristics in Fog Environment. In Recent Trends in Electronics and Communication (pp. 857-863). Springer, Singapore.

[89] POTU, N., BHUKYA, S., JATOTH, C., & PARVATANENI, P. (2022). Quality-aware energy efficient scheduling model for fog computing comprised IoT network. Computers & Electrical Engineering, 97, 107603.