Classification and Performance Study of Task Scheduling Algorithms in Cloud Computing Environment

Najib A. Kofahi
Department of Computer Science, Yarmouk University, Irbid, Jordan
E-mail: nkofahi@yu.edu.jo

Abstract
Cloud computing is becoming very common in recent years and is growing rapidly due to its attractive benefits and features such as resource pooling, accessibility, availability, scalability, reliability, cost saving, security, flexibility, on-demand services, pay-per-use services, use from anywhere, quality of service, resilience, etc. With this rapid growth of cloud computing, there may exist too many users that require services or need to execute their tasks simultaneously by resources provided by service providers. To get these services with the best performance, and minimum cost, response time, makespan, effective use of resources, etc. an intelligent and efficient task scheduling technique is required and considered as one of the main and essential issues in the cloud computing environment. It is necessary for allocating tasks to the proper cloud resources and optimizing the overall system performance. To this end, researchers put huge efforts to develop several classes of scheduling algorithms to be suitable for the various computing environments and to satisfy the needs of the various types of individuals and organizations. This research article provides a classification of proposed scheduling strategies and developed algorithms in cloud computing environment along with the evaluation of their performance. A comparison of the performance of these algorithms with existing ones is also given. Additionally, the future research work in the reviewed articles (if available) is also pointed out. This research work includes a review of 88 task scheduling algorithms in cloud computing environment distributed over the seven scheduling classes suggested in this study. Each article deals with a novel scheduling technique and the performance improvement it introduces compared with previously existing task scheduling algorithms.
Keywords: Cloud computing, Task scheduling, Load balancing, Makespan, Energy-aware, Turnaround time, Response time, Cost of task, QoS, Multi-objective.
DOI: 10.7176/IKM/12-5-03
Publication date: September 30th 2022

1. Introduction

In the last few decades, it is noticed the increased interest in the Internet and information technology (IT) which became an important part of human daily life. One recent and important development in the IT sector is the Cloud Computing (CC). The increased interest in CC by individuals of various affiliations, companies and organizations of various scale has also changed the world of Internet computing and information and communication systems (ICTs) into a new era of computing. CC has emerged in the 1960s as a common computing model with huge data and information storage capability as well as large processing power and different types of services using clusters of commodity computer systems. Also, it is noticed that since the emergence of CC, the interest in cloud services have been increasing every day. This interest is on both sides of the cloud; those who provide the services (Service Providers), and the users (customers) (Rimal, et al., 2009), (Kofahi, et al., 2019).

Cloud computing environment provide on-demand flexible infrastructure and profitable services like storage facility, computing power, and application software, etc. It saves time, money, and effort. CC is rapidly emerging in different types of applications including IT, educational applications, business applications, governance applications, art applications, data storage and backup applications, social networking applications, mailing services, image processing, video streaming, hosting services; etc. (Youssef, 2012) (Baniwal, 2013), (Ray, 2017).

The cloud computing model is essentially dynamic in nature from both the service users and from the service provider viewpoints (Piparo, 2020). From the service provider point of view, resources are updated dynamically, modifying service needs and pricing models accordingly.

Another useful view of cloud computing environment as an internet-based utility that provide low-cost, high-quality information and services based on pay-as -you-go strategy in which guarantees are offered to customers by the cloud service providers through customized Service-Level Agreement (SLA). In cloud computing, users can have access to huge computing power, huge storage, and large variety of application software with very low infrastructure on the side of customers. In the context of cloud computing, SLA do exist between cloud customers and service providers to define the level of service expected by users of the cloud computing environment from the service providers, specifying the metrics by which that service cost is calculated. Also, SLA specifies consequences if service levels expected by customers are not achieved.

(Hofman & Roubtsova, 2020).

According to the National Institute of Standards and Technology (NIST) (Mell, 2011), cloud computing is considered as an evolving model. NIST defines cloud computing as "a model through which a service requester can access, through the cloud network, a pool of computing resources such as (storage, servers, applications, and services) that are supplied by service providers, from anywhere and on-demand, with minimal waiting time and effort". Putting it in another way, cloud computing is defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and other services) that can be rapidly available and released with minimal management effort or service provider interaction" (Zhang, et al., 2010). While the literature contains many definitions of cloud computing, recent research seems to have converged on the definition proposed by NIST. It includes five essential characteristics, three service models, and four deployment models.

According to the eleventh annual Flexera 2022 State of the Cloud Report (Flexera™, 2022), previously known as (RightScale State of the Cloud Report), to compete in today's game, organizations must have the cloud play an important role in their strategy. According to the report, the cloud helps organizations scale, be more active, increase revenue and achieve business goals. Cloud adoption has been accelerated by COVID-19 pandemic, but as a post-pandemic world begins to take shape, new trends in cloud usage are coming into focus.

The **Flexera 2022** State of the Cloud Report survey revealed the experiences and insights of **753** global cloud decision-makers and users about the cloud computing market including public, private, and multi-clouds. The largest groups of respondents come from tech and financial services, followed closely by software and hardware. Also, this year survey includes larger representation from consumer products and services and transportation and logistics. The other category includes industries that represent less than three percent of respondents.

According to the report, public cloud adoption continues to speed up. 8% of all respondents spend more than $60 million, and more than half spend over $2.4 million on public cloud each year. Public cloud spend is even more significant among larger organizations. 37% of enterprises said their annual spend exceeded $12 million and 80% reported that cloud spend exceeds $1.2 million per year. According to the survey, cloud spends by SMBs (Small to Midsized Businesses with fewer than 1,000 employees) reflects a massive acceptance, with 53% of SMBs spending more than $1.2 million—up from 38% reported last year. The report also indicated that the top challenges for organizations today are security, managing cloud spend and a lack of resources or expertise. Multi-cloud strategies remain dominant. Hybrid approaches are adopted by approximately four out of five respondents, with the most common among enterprises a mix of various public and multiple private clouds. (Flexera™, 2022).

Amazon Web Services (AWS), Microsoft Azure (referred to as **Azure)**, and **Google** cloud are the top three public cloud providers according to the report. Also, according to the report, Azure is narrowing the gap with AWS in both the percentage of enterprises using it and the number of virtual machines (VMs) enterprises are running in it. 53% of enterprise AWS users spend at least $1.2 million annually versus 48% for Azure according to the report. Among the three cloud providers. 32% of enterprise Google users spend $1.2 million or more annually.

To be able to understand the task scheduling problem in cloud computing environment, it is useful first to have a closer look at this environment and understand how it works. There are different cloud computing technologies based on the services they provide and the users' view.

The remainder of the paper is organized as follows: Section 2 presents a brief review of the three main cloud computing service models. In Section 3, the task scheduling problem is discussed. The related work about classification of task scheduling techniques in cloud computing environment is discussed in Section 4. In Section 5, the classification of task scheduling algorithms in cloud computing environment suggested in this research work is presented. Section 6 gives a summary and the conclusions of the study.

2. Cloud computing Service Models

In cloud computing environment, services such as processing power, running applications such as zoom, running programs, using storage space for user data and accessing many other services, all of this and other activities is carried out over the internet network instead of having to pay the cost of buying all these resources (Sharma, et al., 2020). Cloud services are provided by service providers in a *pay-as-you-go* manner also referred to as *pay-per-use-demand*.

According to NIST, cloud-computing suppliers offer their resources according to three standard models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Kofahi & Alsmadi, 2018), (Pandey & Farik, 2015). Figure 1 (Wikipedia, n.d.), (Chen, et al., 2020), portrayed these models. These models need not be related. For example, it is possible to provide SaaS implemented on physical machines, without using underlying PaaS or IaaS models. Also, one can run a program on IaaS and access it directly, without using other models.

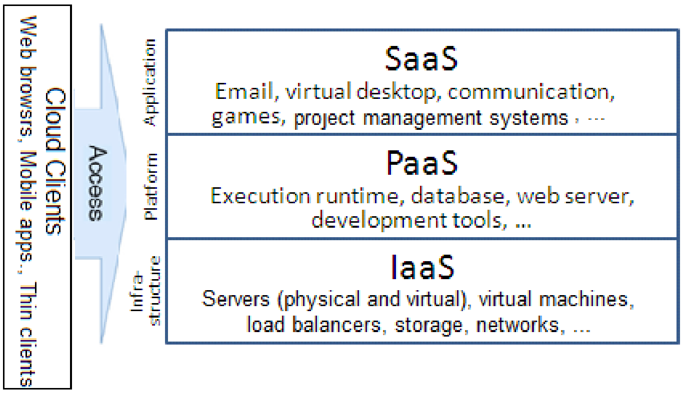

Figure 1. The cloud services (Wikipedia, n.d.), (Chen, et al., 2020)

2.1 Software as a Service (SaaS)
In this model, services are based on pay-per-use manner. The service provider (SP) is responsible for deploying applications and useful software on the cloud. The cloud customers can access and use these services through the Internet using any web browser such as Google. In this model the customers can only use and run these services installed in the cloud, they cannot install or deploy their own applications and software. The cloud provider is the one who is responsible for these services and the cloud infrastructure. Examples on SaaS are Amazon web services which include several dozens of services, online mail with spam protection, project management systems and many other services.

2.2 Platform as a Service (PaaS)
In this model the cloud users can install and deploy their own applications and software on the cloud using the tools provided by the cloud service provider. Cloud users can also control their own applications, where the cloud infrastructure is controlled and managed by the cloud provider. Examples of PaaS are Databases and web servers.

2.3 Infrastructure as a Service (IaaS)
In this model the cloud customers can deploy their own applications in addition to their own operating system on the cloud by renting a cloud infrastructure from the cloud provider, and the customers in this model have the whole controlling on the cloud infrastructure such as Amazon EC2. Examples of IaaS are storage and networks. Examples of IaaS providers are Amazon web services (AWS), Microsoft Windows Azure, Google Compute Engine (GCE), Cisco Metacloud, etc.

Having these services provided to users in a pay-as-you-go manner means that these services should be perfectly utilized, and in an efficient manner to maximize the service providers income and minimize the overall cost incurred by customers and maximizes user satisfaction, through fast and reliable processing of tasks through highly efficient and effective task scheduling techniques. With the new advancements in cloud computing, data access and resource usage from the customer side becomes easier and cheaper because of its zero-cost infrastructure, and as all these needs will be available in one place reachable by all cloud users around the world

3. The Task Scheduling Problem
The general task (or job) scheduling problem has been around since the early days of the computer era. This problem is a combinatorial optimization problem and is one of the critical issues to be solved in the field of computer science as well as in cloud computing. Scheduling problem is already known and proved in computer science to be as an NP-Hard problem. It means that we do not know of any efficient (i.e., polynomial-time) algorithm to solve it.

In cloud computing environment, scheduling problem is related to two types of clients, cloud consumers (users) and cloud service providers (CSPs). Cloud consumers would like to execute their tasks with minimum cost and minimum processing time (or makespan). Cloud service providers will offer resources required to perform services for consumer tasks in an efficient way. Cloud consumer will benefit by selecting the appropriate resources wisely while cloud service provider benefit by maximizing resource utilization and profit.

Cloud computing environment includes huge number of tasks that needs access to many resources and services, shared among cloud customers. In this regard, a serious attention from the research community in cloud computing must be paid to the problem of tasks scheduling, in addition to other issues like resource allocation, resource sharing, load balancing, quality of service (QoS), power consumption, privacy and security, virtual machines, etc. (Pol & Singh, 2021), (Hosseinzadeh, et al., 2020), (Qadir & Ravi, 2020), (Thaman & Singh, 2016), (Awan & Shah, 2015).

Figure 2 (Ahari, et al., 2019) depicts the scheduling of a set of tasks in cloud computing environment. In this figure, customers submit their tasks to the cloud, and the task scheduling algorithm schedule these tasks to the most appropriate virtual machines running in the hosts at the datacenters in an optimal way. The selection of a suitable VM to accommodate the many requests is considered as a challenging research issue in cloud computing. Datacenters represents the infrastructure or hardware consisting of all physical servers. Physical servers are converted into VMs using the virtualization technique and users' jobs are going to run on these VMs in scheduling tasks. VMs are managed by the VMs manager.

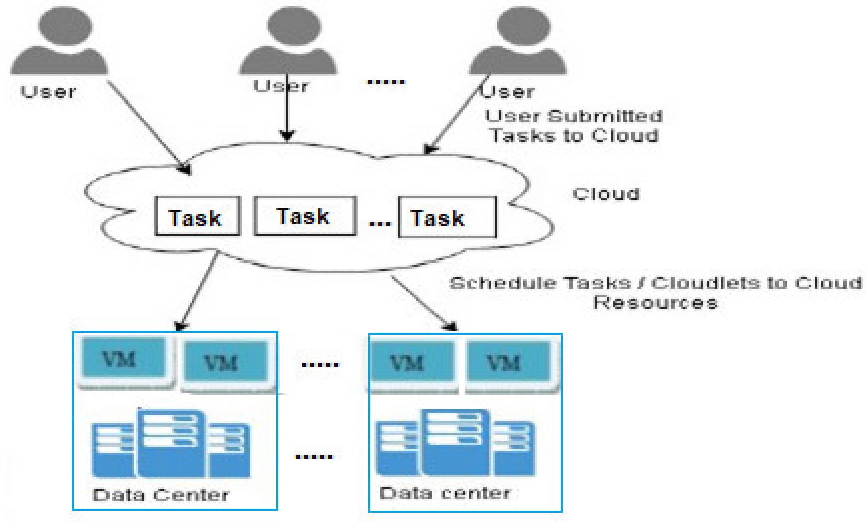

Figure 2. Scheduling process in the cloud (Ahari, et al., 2019)

4. Related Work

Task scheduling algorithms in cloud computing environment is considered as a hot topic of research. It is considered as a major challenge and an essential factor that controls the performance criteria (e.g., execution time, waiting time, bandwidth, network, storage, service cost, makespan, response time, etc.) for all tasks. These performance criteria may differ for different tasks. Consequently, task scheduling becomes a critical component in any efficient cloud computing environment. Other major issues that are of great importance in task scheduling include load balancing, reliability, energy consumption, performance, and dynamic re-allocation of resources to the computing nodes, etc. (Arunarani, et al., 2019). Several research efforts have been conducted concerning task scheduling techniques in cloud computing environment. Several techniques have been used by researchers to develop optimal and near optimal task scheduling algorithms in cloud computing. Among the promising techniques, which are also used to solve other NP-Hard problems are: heuristics, metaheuristics, greedy, and genetic approaches. These techniques are also applied to task scheduling in parallel and distributed systems (Thaman & Singh, 2016),

(Arora & Banyal, 2022) presented a review of the hybrid scheduling algorithms in the cloud computing environment. Hybrid here refers to the combination of two or more techniques. It is noticed that hybridization becomes a trend in recent research work in task scheduling algorithms development in cloud computing environment. The advantage behind the hybridization of different techniques is to take useful features of all algorithms used. The authors also give a classification of the hybrid algorithms and give an analyzes of their objectives, and discuss QoS parameters, and future directions for hybrid scheduling algorithms.

(Pol & Singh, 2021) divided task scheduling algorithms into two main groups: Static Task Scheduling and Dynamic Task Scheduling. The authors also listed the following scheduling methods that are used in cloud computing environment: First Come First Serve (FCFS), Multi-Objective Task Scheduling Algorithms, Multilevel Priority-Based Task Scheduling, Load Balancing Task Scheduling Algorithms, Particle Swarm Optimization based Task Scheduling, Energy-Efficiency based Task Scheduling Algorithm, Cuckoo Optimization based Task Scheduling Algorithm, Fault-Tolerant Workflow Scheduling Algorithm, and Green Energy-Efficient based Task Scheduling Algorithm.

(Ibrahim, 2021) offered a review of task scheduling algorithms in cloud computing. The authors have studied and analyzed several previous task scheduling algorithms in the cloud computing environment used by cloud researchers. The authors pointed out many challenging schedule issues in cloud computing infrastructure like computation time, load balancing, resource utilization, cost, energy consumption, QoS, etc. Many authors applied different parameters like completion time, throughput, and cost to evaluate the system. The authors indicated that job scheduling can be classified into the following algorithm groups: Batch Mode Heuristic Scheduling Algorithms (BMHA), Online Mode Heuristic Algorithms, and Dependency mode heuristic Algorithms.

(Zhang & Abnoosian, 2020) have discussed a methodological analysis of the current state-of-the art mechanisms and techniques used in scheduling in cloud computing environment. The paper also presents a detailed study of the state-of-the-art algorithms in the cloud and directions for future research. The authors classified scheduling mechanisms into deterministic and non-deterministic algorithms. They indicated that it can also be classified into different classes.

(Amini Motlagh, et al., 2020) have expounded a systematic literature review (SLR)-based analysis on the task scheduling approaches. The authors classified task scheduling approaches into three categories, namely: (a) single cloud environments that evaluate cost-aware, energy-aware, multi-objective, and QoS-aware approaches in task scheduling; (b) multicloud environment that evaluates cost-aware, multi-objective, and QoS-aware task scheduling; and (c) mobile cloud environment that is energy-aware and QoS-aware task scheduling. Also, the authors provide analytical discussions to illustrate the advantages and limitations of the existing approaches.

(Bulchandani, et al., 2020) provided a survey on different models of scheduling algorithms in cloud computing environment. The authors have also made a brief comparative study and tabulation of some latest algorithms. They indicated that various algorithms based on different parameters give different performance. Also, they suggest that for a scheduling algorithm to be ideal, different parameters should be considered sequentially after completing some prioritized parameters. The authors classified scheduling algorithms into Static and Dynamic. Static scheduling algorithms are categorized into guided random search based and heuristic based. Genetic algorithm is an example on guided random search based which finds near optimal solution in large solution spaces. In contrast, heuristics-based algorithms make realistic decisions and give reasonable cost or other resource constraint. The authors classified dynamic scheduling into two types: Batch mode and Online mode.

(Sharma, et al., 2020) have given a comparative study of different Task scheduling algorithms in cloud computing environment. The authors classified scheduling algorithms in cloud computing environment into three main groups: Batch Mode Heuristics Algorithms (BMHA), On-line Mode Heuristics Algorithms (OMHA), and Dependency Mode Heuristics Algorithms (DMHA). They also classified task scheduling into the following six types: Cloud Service Scheduling, User-Level Scheduling, Static and Dynamic Scheduling, Heuristic Scheduling, Real Time Scheduling, and Workflow Scheduling. The authors concluded that new algorithms need to be developed that can provide better efficiency and performance.

(Arunarani, et al., 2019) have offered categorization of different strategies of task scheduling algorithms and related evaluation criteria used in cloud computing environments. In the article, the authors considered important aspects related to scheduling strategies and their limitations that may be considered by researchers in the future work. The authors categorize the articles discussed in their paper into three classes: Categorization based on techniques, Categorization based on application, and Categorization based on parameter measure used. Also, the authors pointed out research issues in cloud computing scheduling as future work.

(Keivani & Tapamo, 2019) in their article, have presented categorization of different types of task scheduling algorithms according to certain criteria. Then, the paper gives a review of several scheduling methods in the cloud computing environment. Finally, the algorithms are presented in a table listing their advantages and disadvantages as well as the scheduling tools and factors used.

(Kumar, et al., 2019) have given a systematic and taxonomic review of resource provisioning and scheduling algorithms in cloud computing. The algorithms they discussed in the article are based on QoS parameters, nature of tasks, techniques used to solve problem, advantages of the algorithm and limitations considering deadline or priority as constraint. The authors categorized scheduling algorithms into static and dynamic scheduling, online and offline (batch) mode scheduling, preemptive and non-preemptive scheduling. The authors classified scheduling schemes in cloud computing into: Heuristic, Meta-heuristic, BAT optimization, and Hybrid scheduling algorithms.

Another survey on task scheduling algorithms in cloud computing environment is presented in (Awan & Shah, 2015). The objective of this work is to provide a comparative analysis of task scheduling algorithms in cloud computing environment, where the resources and assets have varying differing cost and performance. The authors classify task scheduling models into eleven different models: Static Scheduling Model, Dynamic Scheduling Model, Hybrid Scheduling Model, Distributed Scheduling Model, Centralized Scheduling Model, Cooperative Scheduling Model, Non-cooperative Scheduling Model, Batch Mode Heuristic Algorithm (BMHA) / off-line Mode, Prompt / On-line Mode, Pre-emptive Scheduling Model, and Non-preemptive Scheduling Model.

5. Classification of Task Scheduling Algorithms in Cloud Computing Environment

In this study we survey and categorize the most popular recent task scheduling algorithms and strategies used in cloud computing environment. They are divided into classes according to certain important performance evaluation metrics that are useful for both the cloud customers and the cloud service providers. In this work we tried to include key performance indicator parameters like response time, makespan time, turnaround time, cost, resource utilization etc. This study includes seven classes that we found of interest for researchers in their recent work in this field. The seven classes are: Load-Balancing Based Task Scheduling Algorithms; Energy-aware Based Algorithms; Execution time or Makespan Based criteria; Turnaround time, Completion time and Response time Performance Metrics; Cost of Task Based Algorithms; Quality of Service (QoS) Based Algorithms; and Dual-Objective & Multi-objective (multi-metrics) task Scheduling. In the following subsections we survey recent research in each class of these categories based on the well-known criteria to be used as evaluation metrics, to be able to compare between task scheduling algorithms developed based on similar criteria. It is important to note that the task scheduling problem is considered as one of the most critical topics in cloud computing since the overall cloud performance is largely depends on it. In general, the performance of task scheduling algorithms is evaluated using several criteria and evaluation metrics including the makespan, energy consumption, resource availability, resource utilization, total execution time (TET), total waiting time (TWT), and total finish time (TFT), the task scheduling cost, minimization of the waiting time of the task, etc. The performance of many task scheduling algorithms is evaluated using several benchmarks and cloud computing evaluation tools (Alshammari, 2018). In the following subsections, the most recent work in each of these categories is being discussed.

5.1 Load-Balancing Based Task Scheduling Algorithms

Load Balancing in cloud computing environment is a technique used to optimize the use of resources and services in the Cloud Computing environment. Load balancing in the cloud environment is one of the important techniques used to make sure that there is a proportionally equal and dynamic distribution of workload over resources and efficient resource utilization. Balancing of the workload distribution leads to higher user and service provider satisfaction and better resource allocation and utilization. Load Balancing in cloud computing environment decreases the delays in data communication between users and datacenters and prevents overloading of datacenters.

(Chukwuneke, et al., 2022) have developed an enhanced hybrid load balancing model in cloud computing environment called (RR PALB) by incorporating Round Robin (RR) load balancing algorithm with the Power Aware Load Balancing (PALB) algorithm. The (RR PALB) model has all nine factors used to consider the load balancing algorithms as success and improved the overall performance. CloudSim toolkit in the JAVA programming language in cloud computing environment was used to implement the (RR PALB) model. The performance of the developed (RR PALB) model was evaluated using Cloud Simulation environment. The (RR PALB) model's accuracy was 99.42 %, compared to 98.64 % for the Round Robin model and 70.77 % for the Power Aware model.

(Mbarek & Mosorov, 2021) have developed a stochastic iterative load balancing algorithm called hybrid Nearest-Neighbor Ant Colony Optimization (ACO-NN) based on Ant Colony Optimization (ACO) metaheuristic and an approximate nearest-neighbor (NN) method. The aim of the research is to establish a dynamic load balancing algorithm for distributed systems to enhance load balancing task management. The ACO-NN algorithm assign each task of the set {T} of tasks to the resource set {N} based on maintaining load balance and increasing the efficiency of the system's task scheduling. Several experiments have been conducted to explore the efficiency of the proposed ACO-NN algorithm. The experimental results of ACO-NN were compared to fourteen heuristic algorithms such as artificial bee colony (ABC), genetic algorithm (GA), simulated annealing (SA), ant colony optimization (ACO), camel herd algorithm (CHA), black hole (BH), greedy randomized adaptive search procedure(GRASP), particle swarm optimization (PSO), traveling salesman problem based on simulated annealing and gene expression programming (TSP-SAGEP), simulated-annealing-based symbiotic organisms search (SOS-SA), multi offspring genetic algorithm (MO-GA), and discrete tree-seed algorithm (DTSA) with their variants (DTSA0, DTSAI, DTSAII).4.1.

The experimental results showed that the hybrid ACO-NN algorithm outperformed existing approaches in maintaining load balance in a dynamic environment.

(Shafiq, et al., 2021) presents a detailed study of load balancing techniques in cloud computing environment. The study includes static, dynamic, and nature-inspired cloud environments, and it covers the Data Center Response Time and overall performance. The authors provide an analysis of the algorithms and point out a research gap for the future research in this field. The authors suggest the following as future research topics: include fault tolerance in the algorithms, task migration, and intelligent algorithms. For future work the authors plan to review nature-inspired and intelligent algorithms such as the application of machine learning or clustering.

(Arora & Dixit, 2020) advanced a hybrid optimization task scheduling algorithm in cloud computing environment, called Elephant Herding-based Grey Wolf Optimizer (EHGWO). The developed algorithm uses load balancing by determining the optimal VMs for executing the reallocated tasks. The EHGWO algorithm uses elephant herding optimization (EHO) and grey wolf optimizer (GWO). Tasks are allocated to the VMs by eliminating the tasks from overloaded VMs by maintaining the system performance. To decide whether the LB must be done or not the load of physical machine (PM), capacity and load of VM is computed. Furthermore, two pick factors, namely, task pick factor (TPF) and VM pick factor (VPF), are considered for choosing the tasks for reallocating them from overloaded VM to underloaded VM. The performance of EHGWO algorithm is evaluated and compared with four other techniques, CMLB, FDLA, EHO, and GWO based on load and makespan. Experiments are implemented in JAVA. Experimental results show that EHGWO algorithm achieves minimum load and minimum makespan with values 0.0221 and 814,264ns, respectively. For future work, the authors plan to analyze the level of LB of EHGWO algorithm.

(Jena, et al., 2020) have developed a new hybrid dynamic load balancing algorithm named as QMPSO to enhance balancing the load among virtual machines in data centers by combining the modified Particle swarm optimization (MPSO) and improved Q-learning algorithm. The main contribution of this hybridization was to enhance the conventional Q-learning algorithm to improve the load balancing by integrating improved Q-learning with MPSO. The efficiency and performance of QMPSO algorithm has been studied in terms of the degree of imbalance, makespan, energy consumption, standard deviation of load, Throughput, The developed QMPSO algorithm was implemented using CloudSim3.0.3 simulation toolkit and its performance was compared with MPSO algorithm, Q-Learning algorithm, and the Improved MPSO (IMPSO) algorithm (Saleh, et al., 2018). The configuration platform used to run the simulator include Intel core i7 processor, 8 GB RAM, 3.4 GHz CPU and Windows 7. Evaluation results revealed that this hybridization resulted in enhancing the performance of datacenters by balancing the load among the VMs by considering the fitness value of each VM. The QMPSO algorithm also improved the makespan, throughput of VMs, energy utilization during load balancing and reduced the waiting time of the tasks effectively as compared with the other three algorithms. For future studies, the authors suggested to carry out load balancing among the dependent tasks dynamically.

(Pradhan & Bisoy, 2020) have developed a Load Balancing algorithm based on Modified PSO task scheduling called (LBMPSO) in cloud computing environment to minimize the overall makespan, maximize resource utilization, and balance the load among the VMs. The proposed algorithm works by using information about the tasks and resources within the datacenters. The developed scheduling algorithm was implemented using CloudSim toolkit and compared with PSO-Based Task Scheduling (PSOBTS), Learning PSO (L-PSO), task-based System Load Balancing Particle Swarm Optimization (TBSLB-PSO), and Dynamic Load Balance *Algorithm* (*DLBA*) techniques. Simulation results revealed that the developed scheduling algorithm outperformed these techniques in reducing makespan and increases the resource utilization. Also, when the number of tasks increases the resource utilization becomes more effective for all cases. For future work, the authors are working to develop a new technique to improve the QoS parameters.

(uz Zaman, et al., 2019) have offered an improved task scheduling heuristic for Large-Scale Computing Systems (LSCSs) that optimizes two objective functions (makespan and power consumption) based on load balancing. In this computational model, each processing element has limited capacity to execute a priori known number of tasks at the same time. A task scheduling heuristic called Extended High to Low Load (ExH2LL) is used that tries to balance the workload over the processing resources. To evaluate this algorithm, a series of simulation experiments were performed. The offered algorithm is compared with five other algorithms: MinMin, High to Low Load (H2LL) heuristic, which is an extension of MinMin, Improved MinMin Task Scheduling (IMMTS), Load Balanced MaxMin (LBM), and M-Level Suffrage-Based Scheduling Algorithm (MSSA). Simulation results showed that ExH2LL outperformed the compared heuristics with respect to makespan, task missing rate, and resource utilization by evenly distributing the load among the available computing resources. However, the authors indicated that the reduction of makespan caused slight increase in power consumption. For future work, the authors intend to consider the effect of reducing the communication cost on makespan, resource utilization, and power consumption. Also, they aim to develop an energy efficient and resource-aware scheduling heuristic that optimizes the makespan and reduces the communication cost.

(Dam, et al., 2018) in their book chapter, proposed a meta-heuristic algorithm for load balancing of virtual machines in cloud computing based on Ant Colony Optimization (ACO). By achieving load balancing, the proposed algorithm optimizes the response time by distributing dynamic workload evenly over the virtual machines. ACO is a computational intelligence (CI) technique, that helps to mimic intelligent behavior from the nature for artifact systems. CloudAnalyst simulation toolkit is used to evaluate the performance of the proposed algorithm. The performance of proposed algorithm is compared with four existing load balancing algorithms: the Genetic Algorithm Gravitational Emulation Local Search (GA-GEL), Genetic Algorithm (GA), Stochastic Hill Climbing (SHC) algorithm, and First Come First Serve (FCFS) algorithm. Comparison results revealed that the proposed algorithm outperformed other algorithms and guarantees the QoS requirement of the customer tasks. For future research, the authors suggest including fault tolerance and job priority in their future work.

The authors in (Samadi, et al., 2018), have enhanced the Heterogeneous Earliest Finish Time (E-HEFT) algorithm that considers the diversity and heterogeneity of virtual machines in a cloud computing cluster (e.g., different bandwidths, transfer rates, and processing capacities). The algorithm uses user-specified financial constraint to get a well-balanced load across the virtual machines while trying to minimize the makespan. CloudSim toolkit package is used to evaluate the performance of E-HEFT algorithm and to compare its performance with the Heterogeneous Earliest Finish Time (HEFT) and MinMin Task Scheduling Heuristic (MinMin-TSH). Simulation results showed that E-HEFT outperformed HEFT and MinMin-TSH algorithms by reducing the makespan and improving load balance between virtual machines. For future work, the authors plan to extend the proposed work to consider other criteria like execution cost, and to improve the algorithm to run in real cloud environments, such as Amazon so that the algorithm can be used with applications in real life environments.

(Mukundha, et al., 2017) have reported in their article a detailed study of various load balancing techniques used in cloud computing environment. The authors also discussed the advantages and disadvantages of the algorithms along with performance metrics. They listed the following objectives of load balancing: increasing the availability of services and user satisfaction, maximizing utilization of resources, decreasing the execution, and waiting times of tasks, improving the performance, maintain system stability, and accommodate future cloud updates. In this article the authors reported a scheduling heuristic based on Particle Swarm Optimization (PSO). The aim is to minimize the total cost of execution of scientific application workflows in Cloud computing environments. In the experiment the authors altered communication cost between resources and the execution cost of computing. The results of the reported approach compared with Best Resource Selection (BRS) heuristic. The results shows that PSO based task scheduling can achieve at least three times cost savings as compared to BRS based mapping. Also, PSO balances the load on compute resources by distributing the tasks properly onto the available resources. As part of our future work, the authors plan to hybridize PSO based heuristic into their workflow management system to schedule workflows of real applications such as brain imaging analysis and others.

(Babu & Samuel, 2016) enhanced the bee colony algorithm for efficient and effective load balancing in cloud environment. The authors used the honeybee’s foraging behavior to balance the load across virtual machines. The tasks removed from overloaded VMs are treated as honeybees and under loaded VMs are the food sources. The enhancement minimized makespan and the number of VM migrations. The performance analysis of the proposed method is carried out in a simulated dynamic environment. Experimental results showed that the proposed algorithm outperformed the traditional bee colony algorithm and minimize makespan and number of migrations and gives better QoS to customers. For future work, the authors suggest enhancing the algorithm with hybridization of other nature inspired algorithms like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc.

(Keshk, et al., 2014) have advanced MACOLB as an improved cloud task scheduling algorithm using Modified Ant Colony Optimization strategy for load balancing. The authors main objective is to balance the system load while trying to minimize the makespan of tasks sets. MACOLB algorithm is used to find the optimal resource allocation for tasks in the dynamic cloud system to minimize the makespan of tasks and increase the performance by balancing the load on resources. The authors used load balancing factor to make the job finishing rate at different resources about the same which improves the load balancing of the schedule. The developed scheduling scheme was simulated for makespan using CloudSim toolkit package. The average makespan of the proposed algorithm was compared with that of the FCFS, Round Robin (RR) and other scheduling algorithms that are based on the basic ACO or Modified Ant Colony Optimization (MACO). Simulation results showed that MACOLB algorithm outperformed MACO, ACO, FCFS and RR algorithms.

For future work, the authors suggested extending MACOLB algorithm with improvements to handle precedence between tasks and costs of resources. Also, they suggest comparing the performance of their algorithm with other metaheuristics strategies.

5.2 Energy-Efficient and Power-Aware Based Algorithms

Nowadays, energy efficiency is considered as one important research topic in cloud computing environment because of the amount of power consumed by the huge number of machines involved in cloud computing environment all over the world. Energy consumption has become one of the main metrics during scheduling decisions and there are many energy-efficient or power-aware scheduling policies suggested by researchers. Several researchers proposed Energy-Aware scheduling algorithms in cloud computing environment to reduce the energy consumption of the underlying cloud infrastructure. In this subsection we present recent energy-aware based task scheduling algorithms in cloud computing environment along with an evaluation of each of these algorithms. Energy consumption is considered an important factor for service providers, because of its environmental and economic consequences. It also affects the user’s cost for using the cloud services.

In their paper, (Ghafari, et al., 2022) offered a comparative analysis of 67 scheduling methods in the cloud system to minimize energy consumption in cloud computing environment during task scheduling. The main idea of this study is to allow concerned people to decide the best scheduling algorithm that optimizes energy properly.

The authors of this article have divided the algorithms into three main classes: heuristic-based task scheduling, metaheuristic-based task scheduling, and other task scheduling algorithms. The authors also discussed the advantages and disadvantages of the proposed algorithms. Future research areas and further advancements in minimizing energy consumption in cloud computing environment are also presented in the article.

(Tuli, et al., 2022) presented a new energy-efficient scheduler for datacenters in cloud computing environment. The technique introduced by the authors is a Gated Graph Convolution Network (GGCN) based holistic resource management called HUNTER. In addition to the efficient utilization of cloud computing servers the proposed technique also decreases thermal hotspots. The introduced technique uses a new cooling specific energy and temperature models, not present in previous approaches. The GGCN based deep surrogate model used by the introduced technique make it possible to quickly generate QoS estimates. The performance of HUNTER has been evaluated through experimentation on simulated and physical cloud computing environments using the CloudSim toolkit and the COSCO framework. The authors used the PADQN, ANN, SDAE-MMQ and HDIC approaches as baselines in their experiments as these are empirically considered to be the best in this type of criteria. Simulation results revealed that HUNTER outperformed existing AI based (HDIC, SDAE-MMQ, ANN and PADQN) and heuristic algorithms (CRUZE and MITEC) based resource schedulers in terms of five performance parameters (temperature, energy consumption, cost, SLA violation and time) by up to 12, 35, 43, 54 and 3 percent respectively. For future work, the authors can extend the proposed technique by factoring in parameters that relate to scalability, security and reliability and their energy consequences. Furthermore, future work may also consider how cooling management can be further enhanced by capturing domain specific methods for cooling.

(Ghafar & Mansouri, 2022) proposed a meta-heuristic energy-aware and cost-efficient Seagull Optimization Algorithm (SOA)-based Task Scheduling called (SOATS) algorithm in cloud computing environment. The proposed SOATS algorithm used the two important features of seagulls, namely migratory and attacking behavior to provide a mathematical model. Seagulls frequently migrate from one place to another place to find plenty of food. Seagulls attack prey when they reach a new place. The performance of the proposed SOATS algorithm is evaluated and compared with four other meta-heuristic algorithms, namely: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Whale Optimization Algorithm (WOA) using MATLAB software environment using Windows 10 Pro operating system platform. Experimental results show that the proposed SOATS algorithm reduce the energy consumption by 31%, 22%, and 28%, and 20% compared with GA, PSO, ACO, and WOA in the case of 500 tasks respectively. Also, the performance of the proposed SOATS algorithm is compared in a heavily loaded environment with three well-known scheduling algorithms, namely: Cost-based Job Scheduling (CJS), Moth Search Algorithm based Differential Evolution (MSDE), and Fuzzy -GA (FUGE). In this case, the proposed SOATS algorithm reduced energy consumption by 27%, 24%, and 23% compared with CJS, FUGE, and MSDE respectively. As future research, the authors plan to combine the proposed algorithm with other meta-heuristic algorithms. Also, the authors plan to consider other cloud computing performance criteria such as security and availability. The authors also want to improve the proposed algorithm using fuzzy theory.

(Samriya, et al., 2021) have introduced a multi-objective algorithm using Emperor Penguin Optimization (EPO) technique to allocate virtual machines in a heterogeneous cloud environment to improve power utilization. The proposed algorithm is analyzed, and its performance is compared with three algorithms: the Binary Gravity Search Algorithm (BGSA), Ant Colony Optimization (ACO)-based algorithm, and Particle Swarm Optimization (PSO)-based algorithm. The performance of the proposed system has been evaluated using CloudSim toolkit with JAVA simulation platform. Experimental results reveled that the proposed EPO-based system has pronounced effect in limiting energy consumption, Service Level Agreement Violation (SLAV), and enlarging QoS requirements in cloud computing environment. The proposed method outperformed the other three methods used in the comparison study. It conserves an average of 55.96% energy and reduces an average of 47.12% SLA violations when compared with the BGSA technique. When compared with ACO, the proposed EPO algorithm conserves an average of 50.29% of energy and reduces an average of 45.27% SLA violations. Comparing simulation results of the EPO algorithm with the PSO, yields the average values for saving energy and SLA reduction of 52.07% and 39.29%, respectively. Thus, the proposed technique decreased the energy consumption of cloud data centers and leads to green computing. For future work, the authors can extend their work with the combination of other methods for better energy efficiency. Moreover, the authors suggest extending the work to consider other objectives like resource wastage of cloud computing VM placement.

(Al-Mahruqi, et al., 2021) have developed a solution for the resource allocation, task scheduling, and optimization of scientific workflows by proposing a Hybrid Heuristic based Energy-efficient cloud Computing service (HH-ECO) algorithm. The developed algorithm guarantees energy-efficient execution by concentrating on executing non-dominant workflow tasks using adaptive mutation and energy-aware migration strategy. The developed HH-ECO algorithm uses the chaotic based particle swarm optimization (C-PSO) principle to achieve optimization of the resource allocation, task scheduling, and resource migration. The CloudSim simulation toolkit is used to evaluate the performance of the developed algorithm. Experimental simulation results show the superiority of the HH-ECO algorithm compared to the existing techniques like the hybrid heuristic workflow scheduling algorithm (HHWS) and distributed dynamic VM management (DDVM). The authors claim that their developed approach improves the optimal makespan to 38.27% and energy conservation to 38.06% compared to the existing techniques. For future work, the authors suggest focusing on VM migration between various data centers as well as the network latency. Also, the authors suggested considering resources such as cooling systems and network racks (e.g., routers and switches) in addition to CPU and RAM as the source of performance and energy consumption.

(Hussain, et al., 2021) have advanced an Energy-aware and Performance-Efficient Task Scheduling (EPETS) Algorithm in a heterogeneous virtualized cloud to solve the issue of energy consumption. There are two stages in this algorithm: the initial scheduling stage helps to reduce execution time and satisfy task deadlines without considering energy consumption, and the second stage task reassignment scheduling to find the best execution location within the deadline limit with less energy consumption. Moreover, to make a reasonable balance between task scheduling and energy saving, the authors suggested an energy-efficient task priority system. The simulation results show that, compared to current energy-efficient scheduling methods of Real coded genetic algorithm (RC-GA), AMTS, and E-PAGA, the proposed solution help to significantly reduce energy consumption and improve performance by 5%-20% with deadline constraint satisfied.

(Kumar & Suman, 2021) proposed a meta-heuristic task scheduling strategy in cloud computing environment called Energy Efficient Black Widow Optimization based Scheduling Algorithm (EEBWOSA) to decrease energy consumption in datacenters and provide better QoS in terms of makespan and cost to cloud customers. The performance of EEBWOSA is tested on workload taken from High-Performance Computing Center North (HPC2N) dataset in Sweden in CloudSim tool. The proposed algorithm reduced energy consumption by 25.69% and 13.52% as compared to GA and PSO techniques. Simulation results also shows that energy consumption can be reduced up to 12% and 13% with better Quality of Service in terms of makespan and cost when compared to GA and PSO. For future work, the authors consider extending the research to minimize the energy consumption at VM migration process. Black Widow Optimization is not trapped at local minima, so, the authors suggests that this property can be promising for new computing paradigm.

(Kollu & Sucharita, 2021) have launched an energy-efficient task scheduling technique in a cloud data centers (DCs) using parallel implementation of hybrid Jaya optimization algorithm (JA). The aim of this model is to decrease energy consumption and maximize the physical resource utilization by considering efficient task scheduling and VM allocation in cloud data centers (DCs). The authors use multi-threading and shared memory parallelism in multicore architectures to develop a parallelized optimization algorithm called Parallel Hybrid Jaya Algorithm" (PHJA). PHJA is based on a combination of Jaya Algorithm (JA) and Genetic Algorithm (GA) in parallel to achieve excellent parallel efficiency performance and reduce the premature convergence time. The authors compared PHJA with four approaches: the Modified Best-Fit Decreasing (MBFD) algorithm, GA, PSO, and Multi-Objective Differential Evolution (MODE) approach. It is found that the PHJA technique decreased the power consumption on average up to 40% over MBFD, 32% over GA, 27% over PSO and 20% over MODE policies. In Future work, the authors plan to develop a new model for considering the quality of services (QoS) of VMs and their correlations and plan to consider interdependencies between the VMs

(Saidi & Hioual, 2020) have researched the issue of reducing power consumption in cloud data centers. The solution proposed by the authors is focusing on task scheduling strategy. They have proposed a two-step new model based on a deep neural network for regression with the multi-criteria decision analysis method, namely: the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method. The strategy selects the best placement for tasks with the minimum energy consumption allocated to the VM by supporting the TOPSIS method. In the first step, a set of tasks arrive in online mode, and arriving tasks are ranked in the queue in descending order according to different parameters. The authors consider d resource types such as CPU, memory, storage, bandwidth, and so on. The tasks arrived in the tasks queue are waiting for scheduling using TOPSIS method. The second step predict the placement of these tasks by building a regression model of energy consumption in the smallest value. The developed task scheduling algorithm is evaluated using PYTHON and the dataset of the individual household electric power consumption (Georges Hebrail and Alice Berard 2012). It automatically obtains the best virtual machine to each task which will select, directly, the minimum energy consumption from regression. Evaluation results show that the general energy consumption in the datacenter is reduced. For future work, the authors suggested testing the developed model in real-time by using the online model according to the nature of this field.

To enhance the performance of the PSO and the Cuckoo Search (CS) techniques, (Mangalampalli, et al., 2020) combined the PSO and CS strategies to develop a hybrid energy aware task scheduling algorithm to solve the cloud task scheduling problem efficiently. User tasks are mapped onto virtual machines in an optimal way based on the priorities of task and VM which is computed based on electricity cost at the datacenter. In this way both energy consumption and makespan are minimized. The CloudSim simulator is used to evaluate the developed algorithm and to compare it with four existing algorithms ACO, GA, PSO and CS. The authors claim that the developed algorithm showed great significance over the existing algorithms in energy consumption and makespan.

(Zong, 2020) propose a Green Cloud Algorithm (GCA) which is a dynamic fusion task-scheduling algorithm intended for green cloud computing, using a hybrid approach based on genetic and ant colony strategies. CloudSim simulator is used to evaluate the algorithm. Experimental results showed that the proposed algorithm considerably reduced both the execution time of tasks and energy consumption of the cloud computing system when compared with GA and ACO approaches and when the number of tasks is large.

(Marahatta, et al., 2019) have developed a dynamic task assignment and scheduling scheme called Energy-aware Fault-tolerant Dynamic Task Scheduling (EFDTS). The purpose of the developed scheme is to optimize the resource utilization, energy consumption and fault tolerance of Cloud Data Centers (CDCs). The performance of the proposed scheme is compared with two existing techniques, namely, the novel Fault-tolerant Elastic Scheduling Algorithm for real-time Tasks in clouds named FESTAL, and the classical Elastic First Fit (EFF) with dynamic resource provisioning mechanism in a virtualized cloud (EFF), which are both constructed for fault tolerance and scheduling schemes. Simulation experiments using the CloudSim toolkit are performed to compare EFDTS with FESTAL and EFF for the task rejection ratio, mean response time, resource utilization, and energy consumption. The experimental results showed that the EFDTS scheme has a lower rejection ratio, shorter response time, higher resource utilization, and lower energy consumption than that of FESTAL and EFF schemes. The EFDTS reduced energy consumption by approximately 14.29% and 35.93% relative to those of FESTAL and EFF, respectively. For future work the authors plan to investigate two issues:

1) How to assign tasks with complementary features in their resource utilization (e.g., computing intensive, memory access intensive, or communication intensive) into physical machines so that resources can be used more efficiently while including the fault tolerant feature.
2) How to predict and use the characteristics of the incoming tasks also required further investigation, as additional energy consumption and even increased energy consumption may be caused by frequent task migration and power switches.

(Zhao, et al., 2019) have advanced a Power-Aware Task Scheduling (PATS) algorithm to reduce energy consumption in heterogeneous cloud computing environment. First, a task scheduling model is developed to predict the power consumption in the cloud environment. This model formulates the task scheduling optimization problem, which tries to minimize the power consumption of the cloud platform. Second, the impact of Physical Machine (PM) type and task execution time on power consumption of cloud platform is analyzed, then a task scheduling algorithm, PATS, is developed to solve the above optimization problem in an efficient way. PATS algorithm is compared with three other algorithms: (1) Cloud Ant Colony Optimization (CACO or Cloud-ACO) algorithm (Ari, et al., 2017), (2) GA (Agarwal & Srivastava, 2016), and (3) the traditional Round Robin (RR) algorithm, in terms of power consumption. CACO is a multi-objective ACO-based task scheduling algorithm. GA is an improved genetic algorithm, aims at reducing the total task execution time of cloud platform by appropriate allocation of the workload to the VMs. In the paper, the PM type selection function of PATS is added to the algorithms to make the comparison algorithms adapt to the cloud environment. Hence, CACO is denoted as PCACO, and the GA under consideration is denoted as PGA. The PATS algorithm is evaluated and compared with the other three scheduling algorithms using CloudSim simulation environment. Simulation results showed that the developed PATS strategy reduced the total power consumption by 23.9%, 50.3%, and 66.6%, compared with PCACO, PGA, and RR respectively. For future work, the authors plan to collect real cloud trace data, and study performance of task scheduling algorithms in large scale cloud platform.

(Marahatta, et al., 2019) have presented an Energy-efficient Dynamic Scheduling scheme (EDS) of real-time tasks for virtualized Cloud Data Centers (CDC). In the EDS scheme, tasks and virtual machines are categorized according to scheduling a historical scheduling record. Then, tasks of the same type are combined and scheduled to exploit the active state of the servers to its maxima. Energy saving during creation and deletion of virtual machines in the EDS algorithm is achieved by using energy efficiencies and optimal operating frequencies of heterogeneous physical hosts. CloudSim simulator is used to evaluate the performance of the EDS algorithm. It is compared with three other algorithms: Energy-aware Task-based Virtual Machine Consolidation (ETVMC) algorithm (Mishra, et al., 2018), Cross Entropy based Vm Placement (CEVP) algorithm (Chen, et al., 2016), and Utilization-based Migration Algorithm (UMA) (Chen, et al., 2015). These three algorithms are also designed for energy optimization. The authors claimed that their experimental results show significant improvement in several parameters including the overall scheduling performance, CDC resource utilization, task guarantee ratio, the mean response time, and energy consumption when compared with the three algorithms ETVMC, CEVP, and UMA. For future research, the authors plan to incorporate fault-tolerance mechanisms by studying how task failure prediction based on machine learning approach, and task scheduling can be combined to achieve better performance.

(Panda & Jana, 2019) have proposed an Energy-efficient Task Scheduling Algorithm (ETSA) that considers the completion time and total utilization of a task on the resources and using a normalization process to make a scheduling decision. The ETSA algorithm is evaluated for energy efficiency and makespan in the heterogeneous environment. Simulation results are compared with six algorithms: random, round robin, dynamic cloud list scheduling (DCLS), energy-aware task consolidation (ETC), energy-conscious task consolidation (ECTC) and MaxUtil. The performance comparison showed that the proposed ETSA algorithm minimizes about 74%, 73%, 78%, 78%, 90% and 84% makespan, and 74%, 74%, 32%, 53%, 86% and 84% energy consumption than random, RR, DCLS, ETC, ECTC and MaxUtil, respectively on average by considering all the instances of the modified benchmark datasets. Simulation results revealed that the proposed ETSA algorithm provides an elegant trade-off between energy efficiency and makespan than the other algorithms. Energy and execution cost can be considered as future work. Also, the authors suggest evaluating the proposed ETSA algorithm by considering a small-scale cluster as future work.

(Fernández-Cerero, et al., 2018) have developed a set of performance and energy-aware strategies for resource allocation, task scheduling, and for the hibernation of virtual machines. The authors used energy and performance-aware scheduling strategies to hibernate the VMs that operate in idle state. The SCORE simulator presented in (Fernández-Cerero, et al., 2018) with modifications is used to evaluate energy efficiency of the proposed model by using an energy-consumption model which considers the following states for each CPU core in a machine: (a) On: 150 W (b) Idle: 70 W. Obtained simulation results showed that a balance between low energy consumption and short makespan can be achieved. Also, experimental results showed that the proposed solution reduced up to 45% of the energy consumption of the CC system. Such significant improvement was achieved by the combination of an energy-aware scheduler with energy-efficiency policies focused on the hibernation of VMs.

(Yadav, et al., 2018) proposed three adaptive algorithms to reduce energy consumption and SLA violation considerably. These algorithms are the Gradient descent-based regression (Gdr), Maximize Correlation Percentage (MCP), and Bandwidth-aware (Bw) selection. The Gdr and MCP are adaptive energy-aware algorithms based on the robust regression model which is used to detect overloaded host. A Bw dynamic VM selection policy select VM based on to network traffic from the overloaded host under SLAs. CloudSim simulator is used to evaluate the efficiency of the proposed algorithms. The obtained results are compared with the values of other algorithms for two metrics namely, the total electric energy consumed by the data center resources and the average percentage of SLA violation in (Beloglazov & Buyya, 2012). Experimental data was provided by the service provider and at different workloads. Simulation results of the proposed algorithms are compared with the algorithms Median Absolute Deviation (MAD), Linear Regression (LR), Inter-Quartile Range (IQR) in (Beloglazov & Buyya, 2012) and AVVMC in (Farahnakian, et al., 2014) and ACSMC in (Ferdaus, et al., 2014). Simulation results from the proposed algorithms showed a significant reduction in energy consumption for overloaded host detection (Gdr, and MCP) with VM selection policy compared with MAD, LR, IQR, AVVMC, and ACSMC. Energy consumption result of the proposed algorithms is 12% less than those of the other algorithms. Thus, the overall energy consumption of the data center is significantly reduced by Gdr, MCP, and Bw. For comparison of VM selection algorithms, Minimum Migration Time (MMT), Maximum Correlation (MC), and Minimum Utilization (MU) in (Beloglazov & Buyya, 2012) are used. The average percentage of SLA violation of the proposed algorithms (GdrBw MCPBw) is significantly less than that of the combined techniques MadMc, MadMmt, MadMu, IqrMc, IqrMmt, LrMc, LrMmt, LrMu, ACSMC, AVVMC, and IqrMu. In future research, the authors plan to study Thermal-aware approach for VM placement. Also, the authors plan to analyze how much energy efficiency, SLA violation improvements and reduction in the operation cost is obtained.

(Gao & Yu, 2017) advanced a two-phase Energy-aware Load Balancing task scheduling algorithm in Heterogeneous Cloud Data Centers called (PLBA). The developed algorithm takes advantage of both dynamic voltage/frequency scaling and virtual machine consolidation to reduce energy consumption in cloud infrastructures. The authors used CloudSim simulation toolkit environment to evaluate the performance of the developed PLBA algorithm and to compare its performance with three other algorithms, namely: Random Algorithm (RA), Round Robin Algorithm (RRA), and Double Threshold Energy Aware Load Balancing Algorithm (DT-PALB) for energy consumption and mean response time. CloudSim simulation results indicated that the proposed PLBA algorithm achieved energy saving of 35.3%, 35.2% and 17.2% compared with the three algorithms RA, RRA, and DT-PALB respectively. For future research, the authors planned to evaluate PLBA in a real cloud data center and to help the cloud providers make better resource management policies. Furthermore, the authors intend to study the effect of other system resources on both energy consumption and average response time.

(Jena, 2017) optimized energy consumption and makespan using an energy efficient Task Scheduling in cloud computing environment based on Clonal Selection Algorithm called (TSCSA). CloudSim-3.0.1 simulator toolkit was used to evaluate the efficiency of the TACSA algorithm. Obtained simulation results were compared with three other algorithms namely: genetic algorithm based TSGA, Maximum Applications Scheduling

Algorithm (MASA) and Random Scheduling Algorithm (RSA). TSCSA algorithm reduced (10-30) % of energy consumption and (5-25) % of time Makespan compared to the other three scheduling algorithms. Simulation results also showed that TSCSA drastically reduced the number of failed tasks, which generally increase the effectiveness of the cloud computing environment. For future research, the authors suggested including more essential objectives like (bandwidth, load balancing, cost etc.) in their optimization model and focusing on more robust algorithm.

(Azad & Navimipour, 2017) considered the importance of both energy consumption and makespan and have proposed a hybrid task scheduling algorithm and priorities in green computing by combining both Ant Colony Optimization (ACO) and cultural algorithm to minimize these two important performance criteria of cloud data centers. The proposed algorithm used the advantages and avoided the disadvantages of both techniques. Evaluation of the proposed algorithm is done using C# language in cloud azure environment. Simulation experiments have been performed on 33.3 GHZ core i5 Intel processor CPU and 4 GB RAM. Experimental results showed that the proposed method outperformed HEFT-upward rank algorithm and ACO in terms of energy consumption and outperforms the HEFT-upward rank in terms of makespan. However, the proposed method improves the makespan but could not excel from ACO. For future work, the authors planned to investigate communication overhead, the voltage/frequency switching overhead, and other uncertain parameters in the actual presence of a heterogeneous environment. Also, the authors plan to use Dynamic Voltage and Frequency Scaling (DVFS) technique to influence on a server's power efficiency.

To reduce energy consumption and makespan of datacenters in the cloud computing environment (Shojafar, et al., 2016) have developed a two-phase energy-efficient genetic-based scheduler, named TETS. In the first phase, tasks are prioritized, and in the second phase tasks are assigned to processing elements. Three prioritization methods are used in the algorithm for prioritizing the tasks and produce optimized initial chromosomes and assign the tasks to an energy-aware model of processors. The developed algorithm was simulated in MATLAB environment. The authors compared the energy consumption and makespan of the developed TETS algorithm with two known algorithms: the Energy-Conscious Scheduling (ECS) (Lee, et al., 2009) and Hybrid GA (Mezmaz, et al., 2011). Simulation results show that TETS reduced energy consumption by 49 % and makespan by 14 %.

(Du, et al., 2014) have presented an energy-efficient task scheduling algorithm that finish all tasks before a deadline for heterogeneous virtual machines in virtualized environments with changeless variable speed. The algorithm is based on combining the list scheduling and the key property of VM speed. The algorithm consists of two key steps: (1) assigning as many tasks as possible to virtual machines with lower energy consumption and (2) keeping the makespan of each virtual machine within the given deadline. The authors have considered the speed of the VMs in their algorithm to achieve optimal solutions for energy consumption. The proposed algorithm is evaluated using the CloudSim toolkit simulation package and is compared with three other excellent algorithms: The heterogeneous earliest finish time (HEFT) algorithm, Genetic algorithm (GA), and Hybrid particle swarm optimization (HPSO) algorithm. Experimental results show that the proposed algorithm outperforms the three scheduling methods by a significant margin in terms of energy consumption.

5.3 Execution time or Makespan Based criteria

The schedule length (Makespan) is the total time taken by the resources to complete the execution of all tasks, or it is the completion time of the last task when all tasks of a schedule have been completed, (i.e., it defines the time from start to finish). It is considered as one of the key metrics for calculating the efficiency of a task scheduling algorithm. Too many scheduling schemes have been developed to achieve the objective of minimizing the makespan criteria in cloud computing environment.

(KashalanYeeser & Arif, 2021) have introduced a modification on the QL_HEFT algorithm (Tong, et al., 2020). In the modified QL_HEFT algorithm, the tasks are distributed between the resources in an efficient way. The WorkflowSim simulator, which is an extended version of the CloudSim is used for the evaluation of the proposed algorithm. Three workflow tests (Montage, Cybershake and Epigenomics) are used to evaluate the experiment. Obtained results have shown that the proposed scheduling algorithm minimizes the makespan of workflow better than the QL_HEFT algorithm. For future work, the authors planned to extend the algorithm to multi-objective optimization and dynamic task scheduling to solve the workflow scheduling problem and consider the financial costs of storage and communication.

(Nalini & Khilar, 2021) have built on the Ant Colony Optimization by introducing a novel Reinforced-Ant Colony Optimization (RACO) algorithm to solve task scheduling problem by combining Ant Colony Optimization (ACO) with Reinforcement learning (RL) and fault tolerance to make the scheduling process fault tolerant, and to achieve minimum makespan. The code of the proposed algorithm and the cloud environment has been implemented in MATLAB 2015 software. Experimental results showed that the proposed RACO algorithm outperformed the standard ACO algorithm and achieved about 60% performance improvement as compared to the standard ACO algorithm. Extra performance improvement of 10% is achieved by introducing the concept of fault tolerance into the proposed RACO algorithm implementation. For future research work, the authors intend to include more objectives including improving the system performance while considering load balancing, QoS parameters and cost minimization objectives. Also, the authors planned for a real-time implementation of the proposed algorithm.

To solve the task scheduling problem, (Samriya & Kumar, 2020) have combined the Fuzzy with the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) and introduced the Fuzzy Topsis (FTOPSIS) approach. The proposed approach minimized makespan using an efficient load balancing strategy by modelling the Whale Optimization Algorithm (WOA) and FTOPSIS. TOPSIS is used to find the best solutions for local optimum, while the whale optimization is used to deal with the constrain in the load balancing which benefits both cloud customers and cloud service providers by improving the cloud computing performance. The proposed approach was evaluated using CloudSim platform. The performance evaluation is done based on makespan, waiting time, cost, and degree of imbalance. For comparison, some of the well-known algorithms are used, namely the GA algorithm, the PSO algorithm, and the swarm based meta-heuristic algorithm, the Artificial Bee Colony (ABC) algorithm. Execution time simulation results showed increasing the throughput of the cloud system by reducing the makespan of the cloud scheduling process. Also, SLA violations is reduced and the QoS is improved. For future work, the authors planned to incorporate in their model additional performance measures such as reliability, security to obtain the trust nodes and the security threats. They also suggested extending the proposed work to be well-suited with independent tasks.

(Attiya, et al., 2020) have solved the job scheduling in cloud computing environment using Harris Hawks optimization (HHO) approach combined with simulated annealing (SA). The developed job scheduling algorithm is called (HHOSA). The SA approach is used as a local search to enhance the convergence rate and improve the quality of solution generated by the HHO algorithm. CloudSim toolkit is used to evaluate the proposed approach along with the standard HHO and other existing state-of-the-art job scheduling algorithms. Simulation results showed that HHOSA significantly outperformed the standard HHO and other existing scheduling algorithms in makespan and achieves an improvement in convergence when search space is large. The authors suggested for future work to extend the algorithm to handle workflow scheduling and energy consumption. Also, the authors claimed that HHOSA can be applied to other optimization problems in fog computing, Internet of things (IoT), feature selection, and image segmentation.

(Kruekaew & Kimpan, 2020) have enhanced the Artificial Bee Colony (ABC) Algorithm to schedule task in homogeneous and heterogeneous cloud computing environments. The proposed algorithm is a combination of Swarm Intelligence algorithm of artificial bee colony and heuristic scheduling algorithm. They called the developed algorithm Heuristic task scheduling with Artificial Bee Colony (HABC). The aim of the developed algorithm was to minimize makespan and balance the loads over the virtual machines. The performance of the HABC algorithm was evaluated using CloudSim simulation toolkit. Experimental results of HABC algorithm were compared for makespan and load balancing with: Ant Colony Optimization (ACO) with standard heuristic algorithm, Particle Swarm Optimization (PSO) with standard heuristic algorithm, and improved PSO (IPSO) with standard heuristic algorithm. The results showed that the HABC with Largest Job First heuristic algorithm (HABC_LJF) outperformed ACO, PSO, and IPSO. For future work, the authors planned (1) to increase the size of datasets, (2) apply the HABC strategy in other practical applications to real-world datasets, and (3) explore the possibility of HABC to deal with multiple job scheduling with different priorities.

(Al-Maytami, et al., 2019) have introduced a novel task scheduling algorithm using Directed Acyclic Graph (DAG) to solve the task scheduling problem in cloud computing environment. The developed algorithm is based on the Prediction of Tasks Computation Time (PTCT) approach. Experiments are run using MATLAB R2013a to evaluate the proposed algorithm and to compare it with four other algorithms: Enhanced Min-Min (Patel, et al., 2015), An Efficient Max-Min (Konjaang, et al., 2016), QoS guided (Han, et al., 2013) and MiM-MaM (Kfatheen & Banu, 2015), developed for heterogeneous systems. The authors claimed that the advanced algorithm provided a significant improvement in tasks makespan, tasks execution time and other parameters and reduced the computation and complexity by using the Principal Components Analysis (PCA) and reduced the Expected Time to Compute (ETC) matrix. For future work the authors planned to go to dynamic scheduling for real-world application graphs and benchmarking in real-world applications. They also planned to focus on improving the total energy utilization and consumption of task scheduling using the PTCT algorithm and comparing their work with relevant state-of-the-art algorithms for cloud energy consumption, such as GreeDi and GreeAODV.

(Mansouri, et al., 2019) have developed a hybrid task scheduling algorithm named Fuzzy Modified Particle Swarm Optimization (FMPSO) that combined Fuzzy system and Modified Particle Swarm Optimization technique to enhance the cloud performance. FMPSO strategy starts with four modified velocity updating methods and roulette wheel selection technique to enhance the global search capability. Then, it uses crossover and mutation operators to overcome some drawbacks of PSO such as local optima. After that, fuzzy inference system is used to calculate fitness. Four input parameters are used for the fuzzy system: CPU speed, RAM size, tasks length, and total execution time. The authors used CloudSim toolkit to evaluate FMPSO algorithm.

Simulation results showed that the proposed algorithm outperformed other approaches in terms of makespan, improvement ratio, imbalance degree, efficiency, and total execution time.

(Gabi, et al., 2018) have developed a task scheduling algorithm based on Orthogonal Taguchi-Based Cat Swarm Optimization (OTB-CSO) to optimize the execution time of tasks in cloud computing environment. In this algorithm, Taguchi orthogonal approach was integrated into tracing mode of Cat Swarm Optimization (CSO) to scheduled tasks on VMs with minimum execution time. CloudSim simulation toolkit was used to implement and evaluate the proposed algorithm based on makespan and degree of imbalance metrics. Experimental results showed that when 20 VMs used, the proposed OTB-CSO was able to minimize makespan of the total tasks scheduled across VMs with 42.86%, 34.57% and 2.58% improvement over minimum and maximum job first (Min– Max), particle swarm optimization with linear descending inertia weight (PSO-LDIW) and hybrid PSO with simulated annealing (HPSO-SA) and returned degree of imbalance with 70.03%, 62.83% and 35.68% improvement over the other algorithms. For future work the authors suggested more study of other computation-based and network-based parameters and the integration of more advanced concepts such as virtual machine migration, energy consumption, multi-objective optimization and to optimize further the algorithm to scale with larger workloads.

(Zohrati, et al., 2018) used a hybrid strategy to enhance dynamic scheduling in cloud computing environment. They combined the Max-min and greedy scheduling strategies. In the proposed algorithm Max-min algorithm is used to sort tasks, then the greedy strategy is used to assign the appropriate VM to the selected task. The load is distributed and balanced over the virtual machines. This means that the number of tasks allocated for each VM may not be the same. CloudSim simulator was used to model and evaluate the proposed technique. The results obtained from the experiments showed that the values of makespan (completion time of the last task) and total waiting time are reduced when compared with Max-min and the greedy algorithms. The proposed approach has improved the waiting time by about 13% and improved makespan value by about 3%. For future work, the authors suggested to enhance the proposed strategy using other parameters such as energy consumption, optimization, and efficiency.

(Saleh, et al., 2018) have introduced an improvement on the particle swarm optimization strategy and developed what is called Improved Particle Swarm Optimization (IPSO) task scheduling algorithm to get an optimal allocation of cloud resources when the number of tasks is large, which is not the case with PSO. In IPSO, the tasks are dynamically divided into groups and each group is assigned to available VM for appropriate resource utilization and in an optimal way. then, the algorithm combines all sub-optimal solutions for all groups into a final allocation map. After that, IPSO attempts to achieve load balancing over the final allocation of resources. IPSO algorithm was compared with three techniques: PSO Ant Colony Optimization (ACO) (Li, et al., 2011), Honeybee based load balancing Algorithm (LBA_HB) (Hashem, et al., 2017), and the standard Round-Robin algorithm. CloudSim is used to evaluate and compare the performance of the IPSO algorithm with the other algorithms in terms of makespan, standard deviation and degree of imbalance. Experimental results show that IPSO algorithm outperforms the other three algorithms. In particular, the proposed algorithm reduced makespan up to 50%. The authors claimed that the load standard deviation and the degree of imbalance are also improved. For future work, the authors suggested extending the IPSO algorithm further by considering other objective functions such as **cost** or **energy consumption**. Also, the authors suggested modifying the IPSO algorithm to work as a parallel workflow scheduling strategy.

(Konjaang, et al., 2017) have introduced the Modified Max-Min (MMax-Min) algorithm to overcome the major problems in task scheduling and resource allocation in Max-Min algorithm. In (MMax-Min), a cloudlet with maximum completion time and minimum completion time is first found and then either of the cloudlets is assigned for execution according to the specifications for the purpose of enhancing the cloud scheduling processes and improving throughput. CloudSim toolkit is used to evaluate the proposed algorithm which is compared with the traditional Max-Min and Round Robin algorithms. The authors found that MMax-Min algorithm outperformed the other two algorithms in terms of makespan and load balancing. For future work, the authors suggested looking at the precedence between cloudlets and resource utilization and to study the energy consumption of the algorithm.

(Sasikaladevi, 2016) Advanced a Minimum Makespan task Scheduling Framework (MMSF) and Minimum Makespan task scheduling Algorithm (MMA) which improved both the total makespan and the virtual machine utilization. The developed MMA is compared with two other scheduling algorithms, First Come First Served (FCFS) and the Shortest Processing time First (SPF) using CloudSchedule in Java as the simulation tool. Experimental results showed that MMA outperformed the two traditional FCFS and SPF algorithms in normal load and heavy load conditions using the total makespan and virtual machine utilization criteria. MMA provided 96% to 100% VM utilization rate. For future work, the authors suggested extending the MMA algorithm further by considering other objectives related to cloud computing such as SLA based scheduling and energy efficient scheduling.

(AL-Sammarraie, et al., 2015) have combined service performance and service cost and developed a hybrid task scheduling algorithm in cloud computing environment called the Performance and Cost Algorithm (PCA). The developed algorithm is based on assigning priorities for tasks according to their profits. The proposed algorithm serves the two objectives: optimizing resource utilization and minimizing makespan, to provide economical and cost competitive services. This way, the service providers offer the best and most efficient services with acceptable cost. The outcome of the developed work is multifaceted that includes minimizing the customers service cost, maximizing the service provider profit, optimize the performance of the services, minimizing the services completion time, and maximizing the resource utilization. PCA is evaluated using a simple implementation of the algorithm using Java language and compared with three algorithms Min-Min, Max-Min, and the Activity Based Costing (ABC) algorithm which is a cost-based scheduling algorithm for the cloud. The makespan of PCA algorithm was more efficient than the other three algorithms in the experiments performed. Also, the average resource utilization of PCA was the best when compared with the other three algorithms (Min-Min, Max-Min and ABC). For future work the authors planned to consider other issues such as temperature resources and energy consumption etc.

(Elzeki, et al., 2012) have studied both the impact of Resource Aware Scheduling Algorithm (RASA) and the atom concept of Max-min strategy in scheduling tasks and presented an improved version of Max-min task scheduling algorithm. RASA task scheduling algorithm is built based on Min-min and Max-min algorithms using their advantages and handling their disadvantages. RASA uses the Min-min strategy to execute small tasks first and applies the Max-min approach to avoid delays in the execution of large tasks. The proposed algorithm support concurrency in the execution of small and large tasks. The developed algorithm is evaluated using JAVA 6 Technology. The authors claimed that evaluation results revealed that tasks scheduling within cloud computing using improved Max-min achieved schedules with comparable lower makespan rather than RASA and original Max-min. In addition, the proposed algorithm produced the same total completion time or smaller than RASA and always smaller than original Max-min. For future work, the authors suggested extending the study by applying the proposed algorithm on actual cloud computing environment and considering many other parameters such as scalability, availability, stability. Also, they suggested optimizing the proposed algorithm and produce more efficient makespan using one of heuristics techniques as genetic algorithm and genetic programming.

5.4 Turnaround time, Completion time and Response time Performance Metrics

In scheduling, turnaround time of a task/job is defined as the total time between the submission of a task/job for processing and the return of the complete output to the customer, that is the time difference between the end of processing & arrival time of the task/job. It is considered as one of the critical measures (metrics) used to evaluate the efficiency of a scheduling strategy. For a given schedule, turnaround time is a performance measure that refers to the total processing time of the schedule until the user gets the complete results. On the other hand, the task/job response time refers to the average time elapsed from submission of the task until the first response is produced and received by the user. Like the turnaround time, the response time is also considered as very important criterion for measuring the system performance and becomes essential for user's QoS. In general, response time directly depends on the availability of resources which is related to the scheduling algorithm used. Several recent studies showed that these performance measures are considered as crucial factor that has the significant effect on the overall performance of cloud computing environment.

(Alhaidari & Balharith, 2021) have proposed a Dynamic Round-Robin Heuristic Algorithm (DRRHA) based on the traditional round-robin algorithm by changing the time slice dynamically based on the mean of the time slice. The remaining burst time of the task is used to decide the continuity of executing the task during the current execution round. CloudSim Plus simulation tool was used to evaluate the proposed algorithm for the waiting time, turnaround time, and response time. Obtained experimental results were compared with four other algorithms namely: the Improved Round Robin CPU scheduling algorithm with Varying time Quantum (IRRVQ) algorithm, the dynamic time slice round-robin algorithm, the improved RR algorithm, and the hybrid task scheduling algorithm combining shortest job first with round Robin schedulers with Dynamic variable task Quantum, named (SRDQ) algorithm. Experimental results showed that the proposed algorithm significantly outperformed the IRRVQ, the dynamic time slice round-robin, the improved RR, and (SRDQ) algorithms, for the three performance measures. For future work the authors suggested the following: (1) enhancing the RR algorithm by finding a new approach for time quantum calculation that combines the dynamic and fixed quantum values to improve the RR algorithm performance, (2) use fuzzy logic and neural networks to predict the best quantum values of tasks automatically, and (3) using the RR algorithm with other meta-heuristic algorithms to improve performance.

(Shi, et al., 2020) have augmented the performance of the classical Min-min task scheduling algorithm through the design of what they called BMin task scheduling algorithm. BMin algorithm first computes the time required for all tasks in the queues waiting to execute on each virtual machine, and then a VM that results in small completion time and gives small workload change is selected. The authors used CloudSim simulation tool in Java to evaluate the proposed algorithm. Experimental results showed that the BMin algorithm reduced task turnaround time, improved throughput, and enhanced the load balance of the resources when compared with the original Min-min algorithm.

In (Hung, et al., 2019), the authors have introduced an improved Max-Min Scheduling Algorithm for load balancing in cloud computing called MMSIA algorithm. The purpose of the MMSIA algorithm was to improve the performance of Max-Min scheduling algorithm. The MMSIA algorithm reduced the completion time of the requests, which in turn may reduce the turnaround time of tasks. It uses machine learning by clustering. The clustering is based on the size of requests and the percentage of utilization of VMs. The algorithm then assigns the largest cluster requests to the VM with the least utilization percent. The cloud environment is emulated using the CloudSim library and programming in the JAVA language. Simulation results showed that MMSIA algorithm has better requests completion time when compared with Max-Min, Min-Min, and Round Robin algorithms. The developed MMSIA algorithm also decreased the processing time in the cloud data center, which results in faster access to cloud services, and improves cloud performance and the quality of service for customers.

(Phi, et al., 2018) have proposed what is called Throttled Modified Algorithm (TMA) based on an algorithm already in place for the purposes of improving the response time of VMs and improving the processing time of tasks in cloud computing environment to enhance the performance and QoS for both the end-users and datacenters. The proposed algorithm performance was evaluated using CloudAnalyts simulation toolkit. Simulation results showed that the response time for users and processing time of cloud datacenters for the proposed algorithm were reduced compared to the two algorithms Round Robin and Throttled algorithm. Also, the proposed algorithm has better load balancing and showed efficiencies when the number of VMs is increased. For future research, the authors planned to study other possible improvements to optimize the performance of the algorithm.

(Gawali & Shinde, 2018) have introduced a heuristic approach to solve the task scheduling and resource allocation problems in cloud computing environment. They used the Modified Analytic Hierarchy Process (MAHP), Bandwidth Aware divisible Task Scheduling (BATS) + BAR optimization, Longest Expected Processing Time (LEPT) preemption, and divide-and-conquer approach. Tasks are processed using MAHP process before their actual allocation to cloud resources. Resource allocation is done using the combined BATS + BAR optimization technique with the bandwidth and the load of the cloud resources as constraints. The designed system used LEPT preemption to preempt resource intensive tasks. The divide-and-conquer approach improves the proposed system performance, as the experimental results showed when compared with the existing BATS and improved differential evolution algorithm (IDEA) frameworks when turnaround time and response time are used as performance metrics. The proposed system was evaluated using the CloudSim simulator environment. The authors compared results obtained from the proposed heuristic with BATS and IDEA frameworks using real CyberShake and Epigenomics as input tasks with respect to turnaround time and response time. Evaluation results showed that the proposed heuristic approach have improved both the turnaround time and response time. The authors claimed that the response time of their proposed heuristic approach is almost 50% less than that obtained using existing BATS and IDEA frameworks. Also, the proposed heuristic approach efficiently allocated resources with high utility and gave the maximum utilization result for CPU, memory, and bandwidth. For future work the authors planned to focus on more scheduling algorithms in which turnaround time and response time will be improved.

(Elmougy, et al., 2017) have combined three task scheduling strategies namely Shortest-Job-First (SJF), Round Robin (RR) scheduling, and the Dynamic variable task Quantum to introduce a hybrid task scheduling algorithm known as (SRDQ) algorithm. The developed algorithm employed the dynamic task quantum to balance waiting time between short and long tasks and split the ready queue into two sub-queues, Q1 for the short tasks and Q2 for the long tasks. Dispatching tasks to resources from Q1 or Q2 were done mutually with two tasks from Q1 and one task from Q2. Three datasets were used to evaluate the performance of two different versions SJF&RR with Dynamic Quantum (SRDQ) and SJF&RR with Static Quantum (SRSQ) with 1,2 and 3 virtual machines using CloudSim simulation toolkit 3.0.3. The performance of the developed algorithms was compared with three other scheduling algorithms: the standard Shortest-Job-First (SJF) algorithm, the standard Round Robin (RR) algorithm and the Time Slice Priority Based RR (TSPBRR) algorithm. Simulation results showed that the developed algorithm outperformed the other three algorithms in minimizing turnaround time, minimizing waiting times, with comparable response time, and to some extent reducing long tasks starvation. For future work, the authors planned to extend their research to find a better task quantum calculation methodology that balance between the static and dynamic quantum values to achieved better reduction in waiting and reducing long tasks starvation.

5.5 Cost of Task Based Algorithms

It is noticed that almost all factor relevant to the study of the effectiveness of task scheduling algorithms in cloud computing environment has been studied in more details except the cost factor, although cost is considered by organizations and cloud customers as an important issue since at the end which matters the most to any business or any cloud users, is how much they must pay for the services they get. To be able to measure the cost, it is necessary to consider some other factors including the cost of resources, execution time, turnaround time, amount of storage used etc. For instance, Amazon EC2 cloud provider established use cost based on the number and the type of VMs used and on the amount of time being used. On the other hand, Google AppEngine cloud provider charges customers based on how much the CPU being used by user's requests. Cloud providers also offer multiple types of VMs with different number of instances, computing capacities, operating system types, and pricing.

(Mansouri & Javidi, 2020) proposed a Cost-based Job Scheduling (CJS) algorithm in cloud computing environment. The proposed algorithm simultaneously considers mixed jobs of data-intensive and computation-intensive types. The proposed algorithm use data, computing power, computation requirement, and network attributes of the cloud datacenters in the job allocation process. CloudSim simulation toolkit was used to evaluate the performance of the proposed CJS algorithm, and its performance was compared with four other existing algorithms: FUGE, Berger, MQS, and HPSO. Simulation results showed that the makespan of CJS was lower by 11% compared to FUGE strategy for 500 jobs. In addition, CJS can increase processor utilization for distributed system while the available and performance QoS requirements are ensured. Also, it is found that the CJS algorithm can decrease the response time of submitted jobs, which may simultaneously consist of data-intensive and computation-intensive jobs when compared with the other four algorithms. For future work, the authors intended to use Taguchi technique, which is based on the mathematical and statistical techniques in an empirical work, for determining weights. Also, the authors planned to investigate the security issue for business clouds. In addition, experiments will be conducted to study the performance of the scheduling strategy in these scenarios using various optimization methods such as genetic algorithm and ant colony optimization.

(Prem Jacob & Pradeep, 2019) have proposed a hybrid task scheduling algorithm in cloud computing environment by combining two optimization techniques namely: the Cuckoo Search (CS) and Particle Swarm Optimization (PSO) strategies. The performance of the proposed CPSO algorithm was evaluated using cloudSim simulation toolkit and was compared with four other algorithms: the Performance and Budget cost Ant Colony Optimization based (PBACO) algorithm, ACO, MIN–MIN, and FCFS algorithms. From the experimental results, the proposed strategy minimized the makespan, cost (all the cost factors such as performance cost and user costs), and deadline violation rate, when compared to the other four algorithms. For future work, the authors will consider the use of the proposed algorithm to optimize other QoS parameters.

(Nasr, et al., 2019) introduced a cost-effective meta-heuristic scheduling algorithm called (CR-AC) using the Chemical Reaction Optimization (CRO) and Ant Colony Optimization (ACO), to solve the workflow scheduling problem with deadline constraint in cloud computing environment. The CR-AC algorithm is implemented using CloudSim simulation toolkit and evaluated using real applications and Amazon EC2 pricing model. The evaluation results of the proposed CR-AC are compared with that of the traditional CRO and ACO algorithms and two other algorithms: modified Particle Swarm Optimization (PSO) (Verma & Kaushal, 2015) and Cost-Effective Genetic Algorithm (CEGA) (Meena, et al., 2016), in terms of scheduling cost or schedule length, makespan, and time complexity. The experimental results revealed that the CR-AC algorithm produced good solutions and achieved better results than the traditional CRO, the ACO, the modified PSO and CEGA algorithms, in terms of the four evaluation measures.

(Alworafi, et al., 2017) presented a Scheduling Cost Approach (SCA) that calculates the cost of using resources like CPU, RAM, bandwidth, and storage used by submitted user tasks. According to the proposed SCA approach, each task is assigned based on task priority considering suitable resources for execution and distribution of load balancing between the VMs in clusters. In SCA, the tasks are distributed between the VMs according to the priority given by users based on the user budget satisfaction. The SCA improved the load balancing by selecting suitable VM for each task. CloudSim simulation toolkit is used to model and evaluate the algorithms. Experimental results of the SCA are compared with the results of FCFS and SJF algorithms which showed that the proposed SCA approach significantly reduced the cost of CPU, RAM, bandwidth, and storage compared to FCFS and SJF algorithms. The comparison of the three approaches were done under same conditions (i.e., task priority and resource cost processing).

(Siwani Sharma & Tyagi, 2017) introduced a Cost-Based task scheduling in cloud computing environment. In the proposed algorithm, the authors divided tasks into different VM's list as high, medium, and low according to their priorities and their need of resources, CPU time and memory. The performance of the proposed algorithm was compared with the Activity Based Costing (ABC) approach which is considered by the authors of this article to be the best technique to schedule cost-based tasks in cloud environment at that time. CloudSim environment with Java implementation was used in evaluating the algorithms. Experimental results showed that the proposed algorithm outperformed the ABC algorithm in terms of performance and cost. For future work, the authors suggested investigating about some other minute parameters in some other framework like Matlab, Simgrid etc.

(Hamad & Omara, 2016) have advanced a Genetic-based Task Scheduling Algorithm called TS-GA in cloud computing environment. The outcome of the developed algorithm was minimizing the total cost of resource usage, minimizing tasks completion time on the VMs, and maximizing resource utilization. The cloud simulator CloudSim simulation toolkit was used to evaluate the proposed algorithm. Simulation results showed that the completion time of all tasks for the proposed algorithm is reduced by (41.83%) and (39.26%) compared with the conventional GA, and RR algorithms, respectively. For the cost parameter, the proposed TS-GA algorithm produced a reduction by (3.6%) and (6.07%) compared with the traditional GA and RR algorithms respectively. The resource utilization of the proposed TS-GA algorithm is improved by (47%) and (30.04%) compared with the traditional GA and RR algorithms, respectively. For future work, the proposed algorithm can be extended to add possibility dynamic characteristic of VMs through run GA. Moreover, more parameters can be added based on the users' requirements.

(Mosleh, et al., 2016) proposed an Adaptive Cost-based Task Scheduling (ACTS) algorithm to provide data access to the virtual machines (VMs) within deadline constraint without increasing the cost. The proposed algorithm fetches data from datacenters in an efficient way and improves the scheduling performance by considering both the completion time of the data access and the data access cost to select the cost-effective path to access data. Completion time of the data access is computed based on the mean and variance of the network service time and the arrival rate of network requests. The task priority is computed based on data access time. Then the costs of data paths are analyzed and allocated based on the task priority. The performance evaluation of the proposed algorithm is carried out using an extended version of the CloudSim simulation environment, and its performance is compared with the cost-efficient task scheduling in (Su, et al., 2013) without considering the cost and completion time of the data access. The comparison is carried out for computation cost, communication cost, execution time, CPU utilization, and bandwidth. Experimental results indicated that the proposed ACTS algorithm achieved better performance than the other algorithm in terms of the performance criteria used in the evaluation process. For future work the authors indicated three issues to be covered in the future research: schedule tasks with undetermined demands by utilizing efficient resource provisioning techniques, computing the cost for regeneration of datasets for exception cases, and the load-balancing for providing efficient cloud computing services.

(Zhao, 2014) developed a cost-aware task scheduling algorithm based on PSO to solve the task scheduling problem in cloud computing environments effectively. The proposed algorithm used a cost-aware fitness function to measure the cost of resource usage, and used the fitness function for time cost, to minimize the processing time and resource usage, and reach a global optimal solution. To evaluate the performance of the proposed modified PSO algorithm, the authors used MATLAB to produce matrices Expected Time Cos (ETC) and Expected Resource Cost (ERC) and used the CloudSim simulation toolkit to simulate the cloud environment. The results of the empirical experiments conducted in a simulated cloud computing environment showed that the modified PSO algorithm minimized the processing time and resource usage when compared with the conventional PSO algorithm. For future work, the authors planned to investigate the effect of data distribution on scheduling, especially for skew data distribution.

(Su, et al., 2013) used two heuristic strategies and developed a cost-efficient task scheduling strategy for running large programs in cloud computing environment. The first heuristic is based on the Pareto dominance concept in which tasks are mapped dynamically to the most cost-efficient VMs. In the second heuristic the monetary costs of non-critical tasks are reduced. The running time and the monetary cost of tasks are considered to map the highest priority tasks to the most efficient VM. The authors evaluated the proposed algorithm using massive numerical experiments on DAGs generated randomly as well as on real applications. The authors compared the performance of the proposed hybrid algorithm with HEFT on large DAGs. HEFT is used as the baseline algorithm for comparison because it is the most frequently used heuristic for the task scheduling problem. Simulation experiments showed that the proposed algorithm resulted in as good makespan as the best-known task-scheduling algorithm and significantly lowered monetary costs.

(Chawla & Bhonsle, 2013) introduced and developed an optimized dynamic cost-based task scheduling algorithm in cloud computing environment based on merging improved cost-based task scheduling (Selvarani & Sadhasivam, 2010) which benefits cloud customers with dynamically optimized resource allocation strategy (Choudhary & Peddoju, 2012) which is beneficial to service providers. The proposed approach also enhanced computation/communication ratio and utilization of available resources by grouping user tasks before allocation of resources. The cloud simulator CloudSim toolkit version3 was used in the evaluation process. Simulation results showed that the processing cost and time are minimum when both task grouping and dynamic optimization of task scheduling were combined.

5.6 Quality of Service (QoS) Based Algorithms

Traditional scheduling algorithms generally aims at completing tasks with the shortest possible time, which in turn improves the overall system throughput and performance. Sometimes the user is not just simply looking for completing task in the shortest completion time. Cloud computing involves users, service providers, and other entities. The goal of different user's QoS may be different among completion time, response time, priority, security policy, processing cost, storage cost, etc. Traditional cloud providers offer at least one class of service with quality of service (QoS) guarantees. The expected QoS of a particular service class with QoS guarantees is defined by specific Service Level Objectives (SLOs) found in the Service Level Agreement (SLA) established between the provider and its customers. The SLA also defines penalties that are applied to providers when SLOs are violated. QoS is the result of research efforts of cloud computing researchers about services performance, cost, makespan, etc., which determines the degree of the satisfaction of cloud customers for the services. (Potluri & Rao, 2017) indicated that the efficiency of the QoS based task scheduling algorithms can be improved by considering three factors: arriving time of the task, time taken by the task to execute on the resource, and the cost in use for the communication. QoS is expressed based on qualitative measures like task completion time, latency, processing cost, throughput, security, reliability, etc. The cloud service provider and cloud customers have different objectives and requirements.

A key issue is how to schedule user tasks efficiently on different resources according to the Quality of Services (QoS) requirements of both cloud computing service providers and customers (Wu, et al., 2013). Because of the vast range within cloud services provided by cloud service providers, from the cloud customer's point of view of an aspect, it is not easy to decide on the ideal cloud service providers to work with and who could fulfill customers QoS requirements relating to desired issues like economics, performance, etc. Therefore, using suitable metrics (parameters) is essential in assessing different available options. QoS metrics play an essential role in selecting cloud service providers and optimizing resource utilization efficiency. (Bardsiri & Hashemi, 2014) discussed various QoS metrics for service providers in cloud computing environment, considering the customer's concern regarding of the huge number of possible cloud solutions. The authors indicated that the performance metrics (parameters) are generally including execution time, response time, throughput, makespan, cost, timeliness (capability to meet deadline when processing a request), etc. The metrics list can be used to help researchers in their future study and is used in the assessment of newly developed algorithms in the field of cloud computing evaluation.

(Tran & Kim, 2021) developed what they called an intelligent and efficient QoS-driven Deep Reinforcement Learning-based (QoS-DRL) cloud task scheduling strategy. The developed strategy focuses on QoS guarantee requirement of a cloud computing environment. The developed scheduler minimized the number of virtual machines required to ensure that all submitted tasks can finish within the specified deadlines. The authors used the 2019 Google cluster dataset to train and evaluate the proposed scheduler through simulation-based experimentation. The dataset included 40 million records which are task events across 12000 machines. Each record consists of over 20 features with useful features for QoS-DRL scheduler implementation which include: start time, end time, CPU rate, maximum CPU rate, assigned memory usage, maximum memory usage, cycles per instruction, memory accesses per instruction. The performance of the designed QoS-DRL outperformed that of the traditional FCFS algorithm. However, the training time of the proposed scheduler was long, and it can be optimized for a better performance. For future work, the authors planned to develop second version of this work, planned to modify the deep neural network architecture, tune system parameters to upgrade our QoS-DRL scheduler performance. The authors also planned to make more comparisons with other heuristics algorithms and evaluate the scheduler with other metrics such as resources utilization, etc.

(Jing, et al., 2021) addressed the problem of the service quality requirements in cloud computing environment and developed a QoS-aware scheduling algorithm called QoS-DPSO, to satisfy the QoS required in cloud computing environment. The authors considered the time, reliability, and cost as a single object problem as a target requirement of QoS requirement. The authors compared the performance of the proposed QoS-DPSO algorithm with four other algorithms: DPSO, PSO, EDF and DBC (Deadline and Budget Constrained) under the same experimental conditions. Experimental results showed that QoS-DPSO revealed an improvement in performance and reliability. Furthermore, simulation experiments also proved the superiority of the performance of the proposed QoS-DPSO algorithm in general and have the advantages of the user task limit time and cost constraints. The authors concluded that, considering the various factors, QoS-DPSO algorithm can provide users with satisfactory services on time constraints and improve the QoS. For future work, the authors will extend the proposed model to DAG task. They will also develop a prediction model and analyze hosts system log to schedule tasks more accurately.

(Ge, Junwei;Yu, Dehua;Fang, Yiqiou, 2021) have developed a QoS based cloud computing task scheduling algorithm by introducing improvements on ant colony technique and using the QoS demand of users and load balancing of cloud platform. The developed algorithm is called BQ-ACS. In their article, the authors first defined a QoS model in terms of two parameters, namely the completion time and execution cost of tasks, and defined a cloud platform load balancing constraint function. Then, to overcome the shortcomings of ant colony algorithm such as falling into local optimum and slow convergence, the authors improved the pheromone update method and expected heuristic function, and the pheromone strength is dynamically changed. To evaluate the proposed algorithm, the CloudSim simulation toolkit is used. The obtained experimental results were compared with the Ant Colony System (ACS) and the Max-Min Ant System (MMAS) algorithms. Experimental results showed that the BQ-ACS algorithm is better than the other two algorithms in terms of user satisfaction (user's service quality) and realizing the load balance of the cloud platform. As a future work, the authors planned to consider the dependencies between tasks and expand the experiments to include more tasks.

(Da Silva, et al., 2020) have devised a new QoS-driven scheduling policy driven by the QoS expected by user tasks from datacenters and service providers. Two main benefits of using the devised QoS-driven strategy are (1) maintaining the QoS of each request as high as possible, considering their QoS requirement and available resources; and (2) promoting fairness by minimizing the variance of the QoS delivered to requests of the same category. The devised policy was evaluated using measurement experiments and simulation fed with traces from a production system and the results were compared with the state-of-the-practice priority-based scheduler. Empirical results showed that the QoS-driven policy devised in this research outperformed the priority-based strategy from the QoS viewpoint. Also, the fairness of the QoS delivered to requests of the same category were much better when using QoS-driven policy. The study showed also the penalties suffered by the service providers due to SLA breaches when the contention level is moderate, leads to a total penalty cost that is about 193% higher, when the priority-based scheduler is used, compared with the total penalty cost due to the use of the QoS-driven scheduler.

(Xue, et al., 2017) proposed a new QoS-based energy-aware task scheduling strategy and the corresponding task scheduling algorithm named QET in heterogeneous cloud computing environment to minimize energy consumption along with satisfying cloud users' quality of service (QoS) through QoS-aware Physical Machine (PM) selection. The performance of the introduced method was evaluated through comprehensive experimental analysis. The authors compared QET performance with two other traditional algorithms namely, Greedy Algorithm and Round Robin (RR) Algorithm by conducting many experiments. From the simulation and experimental results, it was found that the proposed algorithm could save more energy than the other two algorithms during task scheduling, and in the effect of QoS, QET is found to be more advantageous. For future research, the authors will continue to improve and expand the proposed algorithm constantly according to the actual cloud environment to make it adapt to a variety of cloud data centers. Also, the authors planned to test obtained theoretical results in the actual cloud platform to verify the effectiveness and superiority of the algorithm.

(Gabi, et al., 2017) introduced a multi-objective task scheduling problem along with the required users' QoS anticipations and a scheduling model related to the problem, The authors then developed a Dynamic Multi-Objective Orthogonal Taguchi Based-Cat (dMOOTC) algorithm to solve the introduced model. The cloud simulator CloudSim toolkit simulation environment is used to implement and evaluate the performance of the developed algorithm. The dMOOTC algorithm is evaluated with the performance measures of execution time, execution cost, and QoS. The performance of the developed algorithm was compared with four other algorithms: standard Cat Swarm Optimization (CSO), Multi-Objective Particle Swarm Optimization (MOPSO), Enhanced Parallel CSO (EPCSO), and Orthogonal Taguchi Based-Cat Algorithm (OTB-CSO). Simulation results showed that the proposed algorithm outperformed the other algorithms in terms of consumers' QoS expectation.

(Zhang & Xu, 2015) proposed a QoS-constraints based min-min task scheduling algorithm in cloud computing environment called (Mul-QoS-Min-Min). The proposed algorithm aims at fair scheduling and is based on Berger model. According to QoS parameter matching, the proposed algorithm can allocate a task with the best matching resources which best meet user's requirement based on the fairness principle of resource allocation. The evaluation of the proposed algorithm was conducted using the cloud simulator CloudSim toolkit. Simulation results revealed that the (Mul-QoS-Min-Min) algorithm outperformed the conventional min-min scheduling algorithm as far as makespan, rate of discarding task and QoS satisfaction.

(Wu, et al., 2013) have developed a QoS-driven based Task Scheduling algorithm called (TS-QoS) in cloud computing environment. Initially, the developed algorithm dynamically computes the priority of tasks according to the task attributes like privilege, the priority expectation to be scheduled, task length and the pending time of task in queue. Then, tasks are sorted by priority. Next, the completion time of each task is evaluated on different services and tasks are scheduled onto services which can complete the tasks in the shortest possible time according to the sorted task queue. The cloud simulator CloudSim toolkit was used to evaluate the performance of the developed algorithm and compared it with Min-Min and Berger Model. Simulation results showed that the developed algorithm resulted in good performance and load balancing by QoS based on both priority and completion time when compared with Min-Min and Berger Model.

5.7 Dual-Objectives and Multi-objectives (multi-metrics) Task Scheduling

(Chhabra, et al., 2022) modeled the cloud task scheduling (CTS) problem for scheduling concurrent bag-of-task (BoT) applications as an optimization problem to minimize both makespan and energy consumption. The authors developed a three stage multi-objective technique for scheduling tasks of multiple parallel applications in cloud computing environment. The developed algorithm is called Opposition learning enabled Whale Particle Swarm Optimization (OWPSO), which combines the standard Whale Optimization Algorithm (WOA) with both the Opposition-Based Learning (OBL) and Particle Swarm Optimization (PSO) searching mechanisms to overcome the shortcomings in the standard WOA. The cloud simulator CloudSim 3.0.3 with Java and JMetal 5.4 meta-heuristic framework for multi-objective optimization (Anon., n.d.), (Nebro, et al., 2015) was used to evaluate the performance of the proposed OWPSO algorithm. Real workloads extracted from real-supercomputing sites (namely CEA-Curie and HPC2N) were used for benchmarking. The performance of the proposed OWPSO algorithm was compared with several baseline scheduling techniques. Simulation results revealed that the proposed OWPSO approach resulted in makespan reduction in the range of 1.68−18.38% (for CEA-Curie workloads), 2.10−24.32% (for HPC2N workloads), and energy consumption reduction in the range of 0.93−14.70% (for CEA-Curie workloads), and 0.73−25.94% (for HPC2N workloads) over several other well-known meta-heuristics. For future work, the authors planned to test their algorithm against other heuristics and meta-heuristic algorithms with different objective functions by executing larger set of BoTs and workflows as workloads in simulated and real-cloud environment.

(Li, et al., 2022) developed a Multi-Objective Optimization task-scheduling algorithm based on Artificial Fish Swarm Algorithm (MOOAFSA) to solve the task-scheduling problem for secured cloud computing environment. At the beginning, the proposed algorithm initializes the fish population through chaotic mapping, which improved the global optimization capability. Then, MOOAFSA used a dynamic step size and field of view, as well as the introduction of adaptive weight factor, which accelerated the convergence and improved optimization accuracy. After that, MOOAFSA used crossovers and mutations, which made it easier to jump out of a local optimum. To evaluate the performance of the proposed algorithm, the authors used CloudSim simulation toolkit environment to simulate the service traffic tasks (STTs) scheduling experiments. Simulation results showed that compared with Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Fish Swarm Algorithm (AFSA), MOOAFSA significantly accelerated the convergence speed and decreased the task-completion time, load balancing and execution cost by 15.62–28.69%, 66.91–75.62% and 32.37–41.31%, respectively. For future work, the authors planned to consider how to combine fault tolerance with the proposed algorithm MOOAFSA to enhance the reliability of STT scheduling for a secured cloud.

(Kruekaew & Kimpan, 2022) have proposed an independent task scheduling strategy in cloud computing environment called MOABCQ using a multi-objective task scheduling Optimization based on the Artificial Bee Colony (ABC) algorithm with a reinforcement learning technique Q-learning algorithm, that improves the performance of the ABC algorithm. The objectives of the proposed MOABCQ approach include optimizing resource utilization and maximizing VMs throughput. Also, the developed algorithm achieved load balancing between VMs based on makespan, cost, and resource utilization. Performance of the proposed MOABCQ technique was evaluated using CloudSim simulation toolkit and compared with six other load balancing and scheduling algorithms, namely: Max-Min, FCFS, HABC_LJF, Q-learning, MOPSO, and MOCS algorithms using three datasets: Random, Google Cloud Jobs (GoCJ), and Synthetic workload. Experimental results revealed that the algorithms used MOABCQ method has better performance than the other algorithms in terms of reduced makespan, reduced cost, reduced degree of imbalance, increased throughput, and average resource utilization. Also, the testing results with all three datasets revealed that the MOABCQ was able to reduce costs more than the other comparison measures. For future work the authors suggested that task scheduling in a multi-cloud, fog cloud, or edge cloud environment can be challenging and interesting work. Also, other machine learning approaches may be applied further. In addition, the proposed method can also be tested in a real-world environment to observe the performance of the MOABCQ approach.

(Krishnadoss, et al., 2021) proposed an efficient hybridized scheduling algorithm, named Cuckoo Crow Search Algorithm (CCSA) that replicates the parasitic behavior of the cuckoo and food gathering habit of the crow bird. The aim of the proposed algorithm is for improvising the task scheduling process. The crow bird always stares at its neighbors looking for a better food source, and sometimes steals its neighbor's food. The CCSA is based on characteristics of these birds and is designed to identify a suitable VM for processing the task. The performance of the proposed CCSA algorithm had been evaluated using Java (jdk 1.6) with the cloud simulator CloudSim toolkit and obtained results are compared with MO-ACO, ACO and MIN-MIN algorithms regarding to cost and execution time (makespan) parameters. The proposed CCSA approach had produced an improvement of 3.14%, 10.70%, and 21.60% for makespan and had reduced the overall cost to the tune of 4.56%, 11.19% and 19.35% when compared with MO-ACO, ACO, Min-Min algorithms respectively when used with 10 VMs. For future work, the authors suggested considering more QoS parameters for evaluating the efficiency of the proposed hybrid CCSA and the same could be evaluated under real time scenarios.

(Emami, 2021) proposed an Enhanced SunFlower Optimization (ESFO) algorithm to improve the performance of task scheduling regarding the energy consumption and makespan in cloud computing environment. The proposed algorithm finds optimal scheduling in a polynomial time. The performance of the proposed ESFO algorithm was evaluated using Cloudsim toolkit and implemented the core processes in

MATLAB release 2017a. The performance of ESFO algorithm is compared with three algorithms: mean grey wolf optimization (MGWO) algorithm (Natesan & Chokkalingam, 2019), hybrid glowworm swarm optimization (HGSO) algorithm (Zhou & Dong, 2018), and the standard sunflower optimization (SFO) algorithm in terms of execution time (makespan) and energy consumption parameters. Experimental results showed that ESFO outperformed the other three algorithms. The amount of improvement in comparison with the best of the other three algorithms is 0.73% and 2.24% respectively in terms of makespan and energy consumption.

(Ababneh, 2021) Proposed a multi-objective Hybrid Grey Wolf and Whale Optimization (HGWWO) algorithms, which combines two strategies: the Grey Wolf Optimization (GWO) and the Whale Optimization Algorithm (WOA), with the purpose of taking the advantages of both approaches to minimize costs, energy consumption, and total execution time needed for task implementation, as well as improving resource usage. The performance of HGWWO approach is evaluated using CloudSim simulation toolkit and compared with GWO and WOA. Experimental results showed that the proposed approach outperformed the standard algorithms GWO and WOA with respect to costs, energy consumption, makespan, use of resources, and degree of imbalance. For future research, the author planned to use evolutionary algorithms such as PSO and GA, to solve the task scheduling problem.

(Pradeep, et al., 2021) have combined two techniques, the Cuckoo search algorithm (CSA) and the Whale Optimization Algorithm (WOA) to develop an efficient hybrid algorithm called (CWOA). The proposed strategy improved two important factors that affect the efficiency of cloud computing environment namely: quality of service and energy consumption. The performance of the proposed algorithm was compared with Ant Colony Optimization (ACO), CSA and WOA. The results showed that the proposed algorithm outperformed the other algorithms with an improvement of 5.62%, 4.36% and 2.27% regarding to makespan, 16.36%, 19.19% and 13.13% with respect to memory utilization and 19.08%, 19.34% and 16.75% regarding energy consumption criteria, respectively.

(Zheng & Wang, 2021) proposed a hybrid Multi-Objective Bat Algorithm (MOBA) to solve the cloud computing resource scheduling problem by utilizing resources in an efficient way through makespan time, degree of imbalance, cost, and throughput. Several techniques were used by the proposed algorithm to improve the algorithm's global search capability, namely: the back-propagation algorithm based on the mean square error, the conjugate gradient method, and the random walk based on lévy flight. To evaluate the developed algorithm, the authors carried out several numerical experiments using MATLAB r2020a software. Simulation results were indicated with the use of graphs and statistical analysis. The performance of MOBA was compared with other meta-heuristic algorithms: including the multi-objective ant colony optimization (MOACO) algorithm, the multi-objective genetic algorithm (MOGA), multi-objective particle swarm optimization (MOPSO) algorithm, and multi-objective cuckoo search optimization (MOCSO) algorithm. Simulation results showed that (MOBA) was superior to the MOACO algorithm, MOGA, MOPSO, and MOCSO in terms of makespan, degree of imbalance, and throughput. The cost of MOBA is also slightly better than MOACO and MOGA. For future research the authors planned to expand the research to include environmental protection and sustainable economic development, such as the dual-carbon economy and green management.

(Pirozmand, et al., 2021) have introduced a two-step genetic based energy and time aware hybrid technique for scheduling tasks in cloud computing environment to reduce energy consumption and makespan. The authors called their introduced algorithm Genetic Algorithm and Energy-Conscious Scheduling (GAECS) Heuristic. The first step in the GAECS heuristic involves prioritizing tasks and generating primary chromosomes. The second step consists of assigning tasks to the processor using the Energy-Conscious Scheduling Heuristic model. The GAECS algorithm was compared with the following algorithms, namely: the Gravitational Scheduling Algorithm (GSA), Artificial Bee Colony (ABC) algorithm, Dragonfly Algorithm (DA), Linear-PSO, Sigmod-PSO, Chaotic-PSO, Simulated-PSO, and Logarithm-PSO. All algorithms including GAECS are implemented and evaluated using MATLAB 2014a. Experimental results showed that the GAECS has a better makespan and energy consumption compared with the other algorithms used in the study.

(Zhang, et al., 2021) Ali have enhanced the Heterogeneous Earliest Finish Time (HEFT) task scheduling algorithm along with using priority-based constraints as ordering rules to advance a multi-objective rule-based Enhanced Heterogeneous Earliest Finish Time (EHEFT-R) task scheduling algorithm. The aim of this work was to optimize three parameters: task execution efficiency, quality of service (QoS) and energy consumption. The authors used priority-based constraints as ordering rules to optimize the quality of the initial solution. The enhanced heterogeneous earliest finish time (HEFT) algorithm was used next to ensure the required performance of the solution space. To study the performance of the proposed algorithm, and for the purpose of comparison with four other algorithms: QL-HEFT, HEFT-D, HEFT-U and CPOP in small-scale experiments, the authors designed mathematical models to evaluate Makespan, energy consumption and QoS. Evaluation results showed that EHEFT-R performance outperformed the other four algorithms in the three evaluation measures. For large-scale environment, EHEFT-R was compared with the Extended Genetic Algorithm (EGA) and Non-dominated Sorting Genetic Algorithm (NSGAII). In this case also, EHEFT-R algorithm achieved much better results than

EGA and NSGAII algorithms.

(Natesan & Chokkalingam, 2020) presented an improved Grey Wolf Optimization Based Task Scheduling technique in Cloud Computing Environment called Performance-Cost Grey Wolf Optimization (PCGWO) algorithm to achieve optimization in the process of task and resource scheduling in cloud computing environment. The developed PCGWO algorithm is compared to three other scheduling algorithms namely: the standard GWO, Min-Min and FCFS algorithms. CloudSim toolkit was used in the evaluation process of the proposed algorithm. Simulation results showed that the developed algorithm outperformed the other three algorithms in terms of QoS that includes task completion time (makespan), cost, and the maximum number of tasks completed within the deadline. For future work, the authors suggested that the PCGWO algorithm can be used, and its performance can be compared with other meta-heuristic techniques.

(Qadir & Ravi, 2020) presented a Dual Objective Task Scheduling (DOTS) algorithm to minimize the makespan and balance the load between the cloud resources in cloud computing environment. The evaluation of load balancing of tasks between cloud resources is done based on the coefficient of variation. The authors used a simple Weighted Sum Method (WSM) to convert the dual objectives of the introduced algorithm into a single score based on famous Multiple-criteria decision-making (MCDM) technique. The authors compared the performance of their developed algorithm with seven other scheduling algorithms: Novel Heuristic Based Task Scheduling (NHBTS), Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-Min, Max-Min and Resource Aware Scheduling Algorithm (RASA) techniques. The authors used three mathematical models in their comparative study. Three performance parameters are evaluated, namely: Makespan, Resource Utilization and Weighted Sum Score. The evaluation results revealed that the proposed DOTS technique outperformed the other seven scheduling algorithms used in this study. Also, DOTS performed exceptionally well on each objective independently.

(Sharma & Bala, 2020) have proposed a multiple credits-based scheduling algorithm employing four real time factors: Task-Length, Task-Priority, Deadline and Cost, as credits and use Modified K-means Clustering technique to classify users' tasks and virtual machines (VMs). The proposed technique eliminated the drawbacks of the common existing task scheduling algorithms and enhanced the performance and efficiency of cloud computing. CloudSim with Java framework have been used to implement and evaluate the proposed algorithm. Simulation results obtained from the proposed algorithm were compared with the existing algorithm (with two credits: task length and task priority). Results showed that the proposed algorithm considerably outperformed the existing algorithm in makespan time, processing time and cost. For future work the authors planned to research on more QoS factors/parameters and effective load balancing techniques to enhance performance and improve efficiency of cloud computing environment.

(Sanaj & Prathap, 2020) introduce an enhancement on the standard Round Robin task scheduling algorithm and developed what they called Enhanced Round Robin (ERR) task scheduling algorithm to improve the performance of the traditional RR without affecting the desired properties of the standard RR. ERR dynamically considers the time slices for each submitted task that enters the queue. Time slices are determined using the Mean Bursting time of all the tasks waiting in the queue. Therefore, ERR was considered as a minor modification of the standard RR. ERR algorithm was evaluated using CloudSim toolkit simulator for the average waiting time for the user tasks submitted to a certain number of cloudlets. Simulation results indicated that the average waiting time for ERR was less than the standard RR. The authors also claimed that the execution time and the residue energy for ERR was better than other algorithms such as ACO, GA, MPA, Min-Min and PSO.

(Thiyeb & Alhomdy, 2020) proposed a Hybrid task scheduling Algorithm in cloud computing environment based on the Min-min and Max-min heuristics. The proposed algorithm is called Hybrid Algorithm based on the Min-min and Max-min heuristics (HAMM). Cloudsim simulator was used to evaluate the proposed algorithm for five performance parameters, namely: makespan, average of resource utilization, load balancing, average of waiting time and concurrent execution between small length tasks and long size tasks. The results showed that the proposed algorithm outperformed the other two algorithms standard Min-Min and standard Max-Min for those parameters. For future work, the authors planned to enhance the proposed algorithm by using heuristic algorithms characteristics in dynamic task scheduling algorithms with artificial intelligence technologies.

(Thanka, et al., 2019) have advanced a hybrid task scheduling algorithm in cloud computing environment based on Artificial Bee colony strategy and Particle Swarm optimization called it ABPS algorithm. The main purpose of ABPS task scheduling algorithm was to minimize makespan, cost, and maximize resource utilization and balance the load. The CloudSim simulation toolkit was used to evaluate the ABPS algorithm and to compare its performance with the traditional ABC and PSO algorithms. Simulation results revealed that ABPS algorithm based on makespan outperformed ABC and PSO algorithms by 22.07% and 28.12%, respectively. When compared with cost, the ABPS algorithm outperformed ABC and PSO algorithms by 32.41% and 44.49% respectively. Based on resource utilization, the ABPS algorithm outperformed ABC and PSO algorithms by 49.37% and 48.88% respectively, and for the degree of imbalance outperformed ABC and PSO algorithms by 16.21% and 20.51% respectively. For future work, the authors suggested adding security features to improve the security of the task in a cloud environment.

(Madni, et al., 2019) have introduced a meta-heuristic multi-objective algorithm to solve the resource allocation problem in IaaS cloud computing environment using cuckoo search optimization strategy. The authors called their algorithm Multi-Objective Cuckoo Search Optimization (MOCSO) algorithm. MOCSO algorithm was evaluated along with four other algorithms: Multi Objective Ant Colony Optimization (MOACO) algorithm, Multi-Objectives Genetic Algorithm (MOGA), Multi-Objective Model-Metric (MOMM) algorithm, and Multi-Objective Particle Swarm optimization (MOPSO) algorithm under the same experimental environment. The algorithms have been implemented and evaluated using CloudSim simulator framework. Experimental results showed that MOCSO algorithm performed better than the other four algorithms and improved the performance in terms of makespan, cost, and resource utilization in cloud computing environment. MOCSO algorithm generates 9.5%, 10.67%, 19.15% and 8.29% improvements in makespan when compared with the MOACO, MOGA, MOMM and MOPSO algorithms respectively. Also, MOCSO algorithm produced 79.44%, 47.60%, 85.76% and 31.22% cost improvements compared with the MOACO, MOGA, MOMM and MOPSO algorithms. Regarding resource utilization, MOCSO algorithm showed 11.82%, 11.22%, 14.27% and 12.17% improvements compared with the MOACO, MOGA, MOMM and MOPSO algorithms respectively. Consequently, customers cost was reduced, the performance of the cloud also was improved, and the profit for cloud providers was increased due to maximizing resource utilization. For future work, the authors suggested that hybridization of cuckoo search (CS) algorithm with other optimized heuristic and meta-heuristic algorithms may also prove to be beneficial for cloud computing environment.

(Al-Rahayfeh, et al., 2019) have introduced a hybrid approach that combined task scheduling with load balancing. The dominant sequence clustering (DSC) algorithm is used for the purpose of task scheduling, and load balancing is achieved using a Weighted Least Connection (WLC) algorithm. First, users' tasks are clustered thru the DSC algorithm, which represents user tasks as graph of one or more clusters. Next, a rank is associated with each task based on Modified Heterogeneous Earliest Finish Time (MHEFT) algorithm. Tasks are then scheduled according to their priorities. Virtual machines (VMs) are then grouped employing a mean shift clustering (MSC) algorithm utilizing kernel functions. Load balancing is achieved thru the WLC algorithm. Tasks are then distributed among resources according to certain parameters like server weight and capacity in addition to client connectivity to the server. The introduced algorithm was evaluated using CloudSim 3.0 simulator environment for four criteria: response time, makespan, resource utilization, and service reliability and compared with the Temporal Task Scheduling Algorithm (TTSA) (Yuan, et al., 2016). Simulation results showed that the proposed system performed better for the performance metrics, particularly in decreasing response time. The proposed algorithm achieves 98% resource utilization, which is higher than that of the TTSA. For future research, the authors planned to incorporate a new intelligent optimization algorithm to choose the task with the highest priority and to use an efficient load balancing algorithm to enhance execution time.

6. Summary and conclusions

Cloud computing environment is customer-oriented technology and considered as one of these technological evolutions in computing models which evolved from parallel computing, distributed computing, and grid computing. It includes huge number of different types of virtualized assets (resources and services) and tasks. Cloud assets and services are deployed by Cloud Service Providers (CSPs) that needs to be accessed efficiently and fairly and shared among cloud computing environment customers.

Since a very large number of users of different affiliations and interests supported by the cloud computing community and due to the wide range of services provided by service providers in the cloud computing environment these days, the issue of task scheduling in the cloud computing environment has become a key concern of researchers all over the world. It is considered as a critical part and one of the main issues in cloud computing environment which affects the overall cloud system performance.

Therefore, to improve the overall system performance of cloud computing environment by service providers, and to satisfy customers' needs and requirements there is always a need for an efficient task-scheduling algorithm. An effective task and workflow scheduling is considered as a crucial factor for task execution in the cloud computing environment. Task scheduling is considered as one of the most important research problems in cloud computing, and there is always a possibility for an improvement, enhancement and modification of previous work and development of new algorithms. There are a vast variety of task scheduling algorithms exists to improve the overall performance of the cloud system.

In this paper several recent reviews and surveys of cloud computing issues and scheduling algorithms in cloud computing environment has been presented. In addition, various task scheduling algorithms in cloud computing environment have been examined for the various performance measures handled by each of the algorithms. It is found that different scheduling algorithms have been developed based on different scheduling parameters and evaluation criteria. Some algorithms are efficient in certain performance parameter(s) and others may be efficient in another area of performance parameter(s). In this study we tried to concentrate on articles with improved performance from various aspects that are of concern to both cloud computing customers and cloud service providers. The articles present existing research that showed enhancement and improvement in various scheduling criteria and parameters including Load balancing; Energy consumption; Time to complete tasks or makespan and execution time; turnaround time, response time; cost of task; quality of service; and multi-objective criteria. For example, the experimental results of one of the selected algorithms in this study were compared to fourteen heuristic algorithms. The findings of this algorithm show that this algorithm outperforms existing approaches in the other algorithms in maintaining load balance in a dynamic environment. fourteen

This research finds a proper classification of efficient and recent task scheduling algorithms in cloud computing environment. Also, the performance measure of these algorithms is studied and compared with other existing algorithms. These algorithms are used to solve the task scheduling problem in cloud computing environment, which is an NP-hard. One major aim of this research work is to summarize the results of important task scheduling algorithms in cloud computing environment and compile several important research works in the field that contribute to facilitate and shorten the time to carry out research work by other researchers by providing them with detailed references in this field in one place.

It is noticed that many of the newly developed algorithms follow the strategy of combining different scheduling approaches and considering more than one input parameter such as security, reliability, cost, priority, makespan, load balancing, etc. to develop a powerful scheduling model in a cloud computing environment. It is also noticed that the new research work tries enhancing the effectiveness of cloud system scheduling by combining task scheduling and virtual machine consolidation schemes. Also, based upon this study, it is noticed that many of the proposed scheduling algorithms does not consider some of key performance parameters due to which enhancement is required in recently developed algorithms to achieve better performance. We believe that further turnaround-time based, and response-time based algorithms need to be developed and at the same time consider developing efficient algorithms that considers other performance measures.

It is noticed that hybridization becomes a trend in recent research work in task scheduling algorithms development in cloud computing environment by considering the useful features of the techniques used in the developed hybridized algorithm, as seen in this research work. Furthermore, hybridization is recommended by many researchers as a future work to improve their developed algorithms. The following are few examples:

- As part of our future work, the authors in (Mukundha, et al., 2017) plan to hybridize PSO based heuristic into their workflow management system to schedule workflows of real applications such as brain imaging analysis and others.
- As **future research**, the authors in (Ghafar & Mansouri, 2022) plan to combine the proposed algorithm with other meta-heuristic algorithms.
- **For future work** the authors in (Alhaidari & Balharith, 2021) suggested the following:
  (1) enhancing the RR algorithm by finding a new approach for time quantum calculation that combines the dynamic and fixed quantum values to improve the RR algorithm performance,
  (2) use fuzzy logic and neural networks to predict the best quantum values of tasks automatically, (3) using the RR algorithm with other meta-heuristic algorithms to improve performance.
- In the **future**, the author in (Ababneh, 2021) suggested to use evolutionary algorithms such as PSO and GA, to solve the task scheduling problem
- For **future work**, (Babu & Samuel, 2016) suggested enhancing the algorithm with **hybridization** of other nature inspired algorithms like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc.
- For **future work**, (Samriya, et al., 2021) the authors suggested extending their work with the combination of other methods for better energy efficiency and to consider other objectives like resource wastage of cloud computing VM placement.
- **For future work**, (Madni, et al., 2019) suggested that hybridization of cuckoo search (CS) algorithm with other optimized heuristic and meta-heuristic algorithms may also prove to be beneficial for cloud computing environment.

By pointing out the performance evaluation and the future work of the surveyed articles, it is hopeful that this survey will give way to more future research directions regarding task scheduling in cloud computing environment and will lead researchers to develop and present more efficient scheduling techniques in cloud computing environment.

Among the many techniques available to evaluate the performance of the developed task scheduling algorithms, it is found that simulation was the most common. Also, CloudSim simulation toolkit environment was the most used in the performance evaluation process.

Acknowledgment
The author would like to acknowledge Yarmouk University, Irbid-Jordan for the support provided to fulfil this research work. This research was performed while the author was on sabbatical leave from Yarmouk University in the academic year 2021 - 2022.

References
Ababneh, J., 2021. A Hybrid Approach Based on Grey Wolf and Whale Optimization Algorithms for Solving Cloud Task Scheduling Problem. *Mathematical Problems in Engineering,* Volume 2021.
Agarwal, M. & Srivastava, G. M. S., 2016. *A genetic algorithm inspired task scheduling in cloud computing.* s.l., IEEE, pp. 364-367.
Ahari, V., Venkatesan, R. & Latha, D. P. P., 2019. *A survey on task scheduling using intelligent water drops algorithm in cloud computing.* s.l., IEEE, pp. 39-45.
Aladwani, T., 2020. Types of task scheduling algorithms in cloud computing environment. *Scheduling Problems-New Applications and Trends.*
Alhaidari, F. & Balharith, T. Z., 2021. Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling. *Computers,* 10(5), p. 63.
Al-Mahruqi, A. A. H., Morison, G., Stewart, B. G. & Athinarayanan, V., 2021. Hybrid heuristic algorithm for better energy optimization and resource utilization in cloud computing. *Wireless Personal Communications,* 118(1), pp. 43-73.
Al-Maytami, B. A. et al., 2019. A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing. *IEEE Access,* Volume 7, pp. 160916-160926.
Al-Rahayfeh, A., Atiewi, S., Abuhussein, A. & Almiani, M., 2019. Novel approach to task scheduling and load balancing using the dominant sequence clustering and mean shift clustering algorithms. *Future Internet,* 11(5), p. 109.
AL-Sammarraie, N., Alrahmawy, M. & Rashad, M., 2015. A Scheduling Algorithm to Enhance the Performance and the Cost of Cloud Services. *International Journal of Intelligent Computing and Information Sciences,* 15(1), pp. 1-14.
Alshammari, D., 2018. *Evaluation of cloud computing modelling tools: simulators and predictive models,* s.l.: University of Glasgow.
Alworafi, M. A., Dhari, A., Al-Hashmi, A. A. & Darem, A. B., 2017. Cost-aware task scheduling in cloud computing environment. *International Journal of Computer Network and Information Security,* 9(5), pp. 52-59.
Amini Motlagh, A., Movaghar, A. & Rahmani, A. M., 2020. Task scheduling mechanisms in cloud computing: A systematic review. *International Journal of Communication Systems,* 33(6), p. e4302.
Anon., n.d. *Welcome to the jMetal 5 Web Site.* [Online] Available at: http://jmetal.github.io/jMetal/ [Accessed 16 June 2022].
Ari, A. A. A. et al., 2017. *Efficient and scalable aco-based task scheduling for green cloud computing environment.* s.l., IEEE, pp. 66-71.
Arora, N. & Banyal, R. K., 2022. Hybrid scheduling algorithms in cloud computing: a review. *International Journal of Electrical & Computer Engineering (2088-8708),* 12(1), pp. 880-895.
Arora, P. & Dixit, A., 2020. An elephant herd grey wolf optimization (EHGWO) algorithm for load balancing in cloud. *International Journal of Pervasive Computing and Communications,* pp. 259-277.
Arunarani, A., Manjula, D. & Sugumaran, V., 2019. Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems,* Volume 91, pp. 407-415.
Attiya, I., Abd Elaziz, M. & Xiong, S., 2020. Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm. *Computational intelligence and neuroscience,* Volume 2020, pp. 1-17.
Awan, M. & Shah, M. A., 2015. A survey on task scheduling algorithms in cloud computing environment. *International Journal of Computer and Information Technology,* 4(2), pp. 441-448.
Azad, P. & Navimipour, N. J., 2017. An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm. *International Journal of Cloud Applications and Computing (IJCAC),* 7(4), pp. 20-40.
Babu, K. R. & Samuel, P., 2016. Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. In: *Innovations in bio-inspired computing and applications.* s.l.:Springer, pp. 67-78.
Baniwal, R., 2013. Applications of cloud computing in different areas. *International Journal of Computer Science & Communication,* 4(2), pp. 174-176.
Bardsiri, A. K. & Hashemi, S. M., 2014. Qos metrics for cloud computing services evaluation. *International Journal of Intelligent Systems and Applications,* 6(12), pp. 27-33.
Beloglazov, A. & Buyya, R., 2012. Optimal online deterministic algorithms and adaptive heuristics for energy

and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience,* 24(13), pp. 1397-1420.

Bulchandani, N. et al., 2020. A Survey On Task Scheduling Algorithms In Cloud Computing. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* 9(1), pp. 460-464.

Chawla, Y. & Bhonsle, M., 2013. Dynamically optimized cost based task scheduling in Cloud Computing. *International Journal of Emerging trends & technology in computer science,* 2(3), pp. 38-42.

Chen, C.-L., Chiang, M.-L. & Lin, C.-B., 2020. The High Performance of a Task Scheduling Algorithm Using Reference Queues for Cloud-Computing Data Centers. *Electronics,* 9(2), p. 371.

Chen, Q. et al., 2015. *Utilization-based VM consolidation scheme for power efficiency in cloud data centers.* s.l., IEEE, pp. 1928-1933.

Chen, X. et al., 2016. CEVP: cross entropy based virtual machine placement for energy optimization in clouds. *The Journal of Supercomputing,* 72(8), pp. 3194-3209.

Chhabra, A., Huang, K.-C., Bacanin, N. & Rashid, T. A., 2022. Optimizing bag of tasks scheduling on cloud data centers using hybrid swarm intelligence meta heuristic. *The Journal of Supercomputing,* 78(7), pp. 9121-9183.

Choudhary, M. & Peddoju, S. K., 2012. A dynamic optimization algorithm for task scheduling in cloud environment. *International Journal of Engineering Research and Applications (IJERA),* 2(3), pp. 2564-2568.

Chukwuneke, C. I. et al., 2022. An Enhanced Load Balancing Model in Cloud Computing Environment. *International Journal of Research Publication and Reviews,,* 3(6), pp. 579-587.

Da Silva, G. F. et al., 2020. QoS-driven scheduling in the cloud. *Journal of Internet Services and Applications,* 11(1), pp. 1-36.

Dam, S., Mandal, G., Dasgupta, K. & Dutta, P., 2018. An ant-colony-based meta-heuristic approach for load balancing in cloud computing. In: *Applied Computational Intelligence and Soft Computing in Engineering.* s.l.:IGI Global, pp. 204-232.

Du, G., He, H. & Meng, Q., 2014. Energy-efficient scheduling for tasks with deadline in virtualized environments. *Mathematical Problems in Engineering,* Volume 2014, pp. 1-7.

Elmougy, S., Sarhan, S. & Joundy, M., 2017. A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud computing,* 6(1), pp. 1-12.

Elzeki, O., Reshad, M. & Elsoud, M. A., 2012. Improved max-min algorithm in cloud computing. *International Journal of Computer Applications,* 50(12), pp. 22-27.

Emami, H., 2021. Cloud task scheduling using enhanced sunflower optimization algorithm. *ICT Express.*

Endo, P. T. et al., 2016. High availability in clouds: systematic review and research challenges. *Journal of Cloud Computing,* 5(1), pp. 1-15.

Farahnakian, F. et al., 2014. Using ant colony system to consolidate VMs for green cloud computing. *IEEE Transactions on Services Computing,* 8(2), pp. 187-198.

Ferdaus, M. H., Murshed, M., Calheiros, R. N. & Buyya, R., 2014. *Virtual machine consolidation in cloud data centers using ACO metaheuristic.* s.l., Springer, pp. 306-317.

Fernández-Cerero, D. et al., 2018. SCORE: Simulator for cloud optimization of resources and energy consumption. *Simulation Modelling Practice and Theory,* Volume 82, pp. 160-173.

Fernández-Cerero, D. et al., 2018. Security supportive energy-aware scheduling and energy policies for cloud environments. *Journal of Parallel and Distributed Computing,* Volume 119, pp. 191-202.

Flexera™, 2022. *Flexera™ State of the Cloud Report: The post-pandemic world comes into focus and FinOps practices gain momentum,* s.l.: s.n.

Gabi, D., Ismail, A. S., Zainal, A. & Zakaria, Z., 2017. *Quality of service (QoS) task scheduling algorithm with Taguchi orthogonal approach for cloud computing environment.* s.l., Springer, pp. 641-649.

Gabi, D. et al., 2018. Orthogonal Taguchi-based cat algorithm for solving task scheduling problem in cloud computing. *Neural Computing and Applications,* 30(6), pp. 1845-1863.

Gao, Y. & Yu, L., 2017. *Energy-aware load balancing in heterogeneous cloud data centers.* s.l., s.n., pp. 80-84.

Gawali, M. B. & Shinde, S. K., 2018. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing,* 7(1), pp. 1-16.

Ge, Junwei;Yu, Dehua;Fang, Yiqiou, 2021. *Multi-dimensional QoS Cloud Computing Task Scheduling Strategy Based on Improved Ant Colony Algorithm.* s.l., IOP Publishing, p. 012031.

Ghafari, R., Kabutarkhani, F. H. & Mansouri, N., 2022. Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. *Cluster Computing,* pp. 1-59.

Ghafar, R. & Mansouri, N., 2022. An Efficient Task Scheduling Based on Seagull Optimization Algorithm for Heterogeneous Cloud Computing Platforms. *International Journal of Engineering,* 35(2), pp. 433-450.

Hamad, S. A. & Omara, F. A., 2016. Genetic-based task scheduling algorithm in cloud computing environment. *International Journal of advanced Computer science and applications,* 7(4), pp. 550-556.

Han, H., Deyui, Q., Zheng, W. & Bin, F., 2013. *A Qos Guided task Scheduling Model in cloud computing environment.* s.l., IEEE, pp. 72-76.

Hashem, W., Nashaat, H. & Rizk, R., 2017. Honey bee based load balancing in cloud computing. *KSII Transactions on Internet and Information Systems (TIIS),* 11(12), pp. 5694-5711.

Hofman, C. & Roubtsova, E., 2020. *A reference model for a service level agreement.* s.l., Springer, pp. 55-68.

Hosseinzadeh, M. et al., 2020. Multi-objective task and workflow scheduling approaches in cloud computing: a comprehensive review. *Journal of Grid Computing,* , 18(3), pp. 327-356.

Hung, T. C., Hieu, L. N., Hy, P. T. & Phi, N. X., 2019. *MMSIA: improved max-min scheduling algorithm for load balancing on cloud computing.* s.l., s.n., pp. 60-64.

Hussain, M. et al., 2021. Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing. *Sustainable Computing: Informatics and Systems,* Volume 30, p. 100517.

Ibrahim, I. M., 2021. Task scheduling algorithms in cloud computing: A review. *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* 12(4), pp. 1041-1053.

Jacob, L. & Jeyakrishnan, V., 2013. Survey on Scheduling Techniques in Cloud Computing. *International Journal of Engineering Research and technology (IJERT),* 2(12), pp. 2979-2982.

Jena, R., 2017. Energy efficient task scheduling in cloud environment. *Energy Procedia,* Volume 141, pp. 222-227.

Jena, U., Das, P. & Kabat, M., 2020. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. *Journal of King Saud University-Computer and Information Sciences,* Volume 34 (2022), pp. 2332-2342.

Jing, W. et al., 2021. QoS-DPSO: QoS-aware task scheduling for cloud computing system. *Journal of Network and Systems Management,* 29(1), pp. 1-29.

KashalanYeeser, Z. & Arif, K. I., 2021. An Efficient tasks scheduling using DAG for Cloud Computing. *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* 12(13), pp. 6841-6857.

Keivani, A. & Tapamo, J.-R., 2019. *Task scheduling in cloud computing: A review.* s.l., IEEE.

Keshk, A. E., El-Sisi, A. B. & Tawfeek, M. A., 2014. Cloud Task Scheduling for Load Balancing based on Intelligent Strategy. *International Journal of Intelligent Systems and Applications,* 6(5), pp. 25-36.

Kfatheen, S. V. & Banu, M. N., 2015. *MiM-MaM: A new task scheduling algorithm for grid environment.* s.l., IEEE, pp. 695-699.

Kofahi, N. A. & Alsmadi, a. T., 2018. A comparative study of Service Brokers scheduling policies in Cloud Computing. *International Journal of Advances in Computer Science & Its Applications – IJCSIA 2018,* 8(2), pp. 69 - 75.

Kofahi, N. A., Alsmadi, T., Barhoush, M. & Moy'awiah, A., 2019. Priority-Based and Optimized Data Center Selection in Cloud Computing. *Arabian Journal for Science and Engineering,* 44(11), pp. 9275 - 9290.

Kollu, A. & Sucharita, V., 2021. Energy efficiency in cloud data centres using parallel hybrid Jaya algorithm. *International Journal of Pervasive Computing and Communications,* 17(3), pp. 315-328.

Konjaang, J., Ayob, F. H. & Muhammed, A., 2017. AN OPTIMIZED MAX-MIN SCHEDULING ALGORITHM IN CLOUD COMPUTING. *Journal of Theoretical & Applied Information Technology,* 95(9), pp. 1916-1922.

Konjaang, J. K., Maipan-Uku, J. & Kubuga, K. K., 2016. An efficient max-min resource allocator and task scheduling algorithm in cloud computing environment. *arXiv preprint arXiv:1611.08864.*

Krishnadoss, P. et al., 2021. CCSA: Hybrid cuckoo crow search algorithm for task scheduling in cloud computing. *International Journal of Intelligent Engineering and Systems,* 14(4), pp. 241-250.

Kruekaew, B. & Kimpan, W., 2020. Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing. *International Journal of Computational Intelligence Systems,* 13(1), pp. 496-510.

Kruekaew, B. & Kimpan, W., 2022. Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning. *IEEE Access,* Volume 10, pp. 17803-17818.

Kumar, M., Sharma, S. C., Goel, A. & Singh, S. P., 2019. A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications,* Volume 143, pp. 1-33.

Kumar, M. & Suman, 2021. *Energy Efficient Scheduling in Cloud Computing using Black Widow Optimization.* Gurgaon, India, IOP Publishing, pp. 1-11.

Lee, Y., Choon, o. & Zomaya, A. Y., 2009. *Minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling.* s.l., IEEE, pp. 92-99.

Li, K. et al., 2011. *Cloud task scheduling based on load balancing ant colony optimization.* Liaoning, China, IEEE, pp. 3-9.

Li, W. et al., 2022. Multi-Objective Optimization of a Task-Scheduling Algorithm for a Secure Cloud. *Information,* 13(2).

Li, Z., O'brien, L., Zhang, H. & Cai, R., 2012. *On a catalogue of metrics for evaluating commercial cloud services.* s.l., IEEE, pp. 164-173.

Madni, S. H. H., Abd Latiff, M. S. & Ali, J., 2019. Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. *Arabian Journal for Science and Engineering,* 44(4), pp. 3585-3602.

Maheswari.R & S.Selvi, 2013. A Survey on Scheduling Algorithms in Cloud Computing. *International Journal of Engineering Research & Technology(IJERT),* 2(10), pp. 3250-3255.

Mangalampalli, S., Mangalampalli, V. K. & Swain, S. K., 2020. Energy Aware Task Scheduling Algorithm In Cloud Computing Using Pso And Cuckoo Search Hybridization. *Solid State Technology,* 63(6), pp. 13995-14010.

Mansouri, N. & Javidi, M. M., 2020. Cost-based job scheduling strategy in cloud computing environments. *Distributed and Parallel Databases,* 38(2), pp. 365-400.

Mansouri, N., Zade, B. M. H. & Javidi, M. M., 2019. Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. *Computers & Industrial Engineering,* Volume 130, pp. 597-633.

Marahatta, A. et al., 2019. Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. *IEEE Transactions on Cloud Computing.*

Marahatta, A. et al., 2019. Energy-aware fault-tolerant dynamic task scheduling scheme for virtualized cloud data centers. *Mobile Networks and Applications,* 24(3), pp. 1063-1077.

Mbarek, F. & Mosorov, V., 2021. Hybrid Nearest-Neighbor Ant Colony Optimization Algorithm for Enhancing Load Balancing Task Management. *Applied Sciences,* 11(22), p. 10807.

Meena, J., Kumar, M. & Vardhan, M., 2016. Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. *IEEE Access,* Volume 4, pp. 5065-5082.

Mell, P. a. G. T., 2011. *The NIST definition of cloud computing,* s.l.: U.S. Department of Commerce.

Mezmaz, M. et al., 2011. A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing,* 71(11), pp. 1497-1508.

Mishra, S. K. et al., 2018. Energy-efficient VM-placement in cloud data center. *Sustainable computing: informatics and systems,* 20(48), pp. 48-55.

Mosleh, M. A., Radhamani, G., Hazber, M. A. & Hasan, S. H., 2016. Adaptive cost-based task scheduling in cloud environment. *Scientific Programming,* Volume 2016, pp. 1-9.

Mukundha, C., Venkatesh, N. & Akshay, K., 2017. A comprehensive study report on load balancing techniques in cloud computing. *International Journal Of Engineering Research And Development e-ISSN,* 13(9), pp. 35-42.

Nalini, J. & Khilar, P., 2021. Reinforced Ant Colony Optimization for Fault Tolerant Task Allocation in Cloud Environments. *Wireless Personal Communications,* 121(4), pp. 2441-2459.

Nasr, A. A., El-Bahnasawy, N. A., Attiya, G. & El-Sayed, A., 2019. Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint. *Arabian Journal for Science and Engineering,* 44(4), pp. 3765-3780.

Natesan, G. & Chokkalingam, A., 2019. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express,* 5(2), pp. 110-114.

Natesan, G. & Chokkalingam, A., 2020. An improved grey wolf optimization algorithm based task scheduling in cloud computing environment. *Int. Arab J. Inf. Technol,* 17(1), pp. 73-81.

Nebro, A. J., Durillo, J. J. & Vergne, M., 2015. *Redesigning the jMetal multi-objective optimization framework.* Madrid, Spain, ACM Press, pp. 1093-1100.

Panda, S. K. & Jana, P. K., 2019. An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. *Cluster Computing,* 22(2), pp. 509-527.

Pandey, S. & Farik, M., 2015. Cloud computing security: latest issues & countermeasures. *Int. J. Sci,* 4(11), pp. 3-6.

Patel, G., Mehta, R. & Bhoi, U., 2015. Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing. *Procedia Computer Science,* Volume 57, pp. 545-553.

Phi, N. X., Tin, C. T., Thu, L. N. K. & Hung, T. C., 2018. Proposed load balancing algorithm to reduce response time and processing time on cloud computing. *Int. J. Comput. Netw. Commun,* 10(3), pp. 87-98.

Piparo, T. L., 2020. The challenges with the service level agreement for the cloud computing buying organization. *Researchgate. Net.*

Pirozmand, P. et al., 2021. Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. *Neural computing and applications,* 33(19), pp. 13075-13088.

Pol, S. S. & Singh, A., 2021. *Task Scheduling Algorithms in Cloud Computing: A Survey.* s.l., IEEE, pp. 244-249.

Potluri, S. & Rao, K. S., 2017. Quality of service based task scheduling algorithms in cloud computing. *International Journal of Electrical and Computer Engineering,* 7(2), pp. 1088-1095.

Pradeep, K. et al., 2021. CWOA: Hybrid Approach for Task Scheduling in Cloud Environment. *The Computer Journal.*

Pradhan, A. & Bisoy, S. K., 2020. A novel load balancing technique for cloud computing platform based on PSO. *ournal of King Saud University-Computer and Information Sciences.*

Prasanth, A., 2012. Cloud computing services: A Survey. *International Journal of Computer Applications,* 46(3), pp. 25-29.

Prem Jacob, T. & Pradeep, K., 2019. A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. *Wireless Personal Communications,* 109(1), pp. 315-331.

Qadir, O. A. & Ravi, D. G., 2020. Dual Objective Task Scheduling Algorithm in Cloud Environment.. *International Journal of Advanced Trends in Computer Science and Engineering,* 9(3), pp. 2527-2534.

Ray, P. P., 2017. An introduction to dew computing: Definition, concept and implications. *IEEE Access,* Volume 6, pp. 723-737.

Rimal, B. P., Choi, E. & Lumb, I., 2009. *A taxonomy and survey of cloud computing systems.* s.l., IEEE, pp. 44-51.

Saidi, K. & Hioual, O., 2020. *Task Scheduling-Energy Efficient in Cloud Computing.* s.l., Springer, pp. 533-540.

Saleh, H., Nashaat, H., Saber, W. & Harb, H. M., 2018. IPSO task scheduling algorithm for large scale data in cloud computing environment. *IEEE Access,* Volume 7, pp. 5412-5420.

Samadi, Y., Zbakh, M. & Tadonki, C., 2018. *E-HEFT: enhancement heterogeneous earliest finish time algorithm for task scheduling based on load balancing in cloud computing.* s.l., IEEE, pp. 601-609.

Samriya, J. K. et al., 2021. Intelligent SLA-Aware VM Allocation and Energy Minimization Approach with EPO Algorithm for Cloud Computing Environment. *Mathematical Problems in Engineering,* Volume 2021, pp. 1-13.

Samriya, J. & Kumar, N., 2020. A QoS Aware FTOPSIS-WOA based task scheduling algorithm with load balancing technique for the cloud computing environment. *Indian Journal of Science and Technology,* 13(35), pp. 3675-3684.

Sanaj, M. & Prathap, P. J., 2020. *An Enhanced Round Robin (ERR) algorithm for Effective and Efficient Task Scheduling in cloud environment.* s.l., IEEE, pp. 107-110.

Sasikaladevi, N., 2016. Minimum makespan task scheduling algorithm in cloud computing. *International Journal of Grid and Distributed Computing,* 9(11), pp. 61-70.

Selvarani, S. & Sadhasivam, G. S., 2010. *Improved cost-based algorithm for task scheduling in cloud computing.* s.l., IEEE.

Shafiq, D. A., Jhanjhi, N. & Abdullah, A., 2021. Load balancing techniques in cloud computing environment: A review. *Journal of King Saud University-Computer and Information Sciences.*

Sharma, P. et al., 2020. A survey on various types of task scheduling algorithm in cloud computing environment. *International Journal Of Scientific & Technology Research,* 9(1), pp. 1513-1521.

Sharma, V. & Bala, M., 2020. An improved task allocation strategy in cloud using modified K-means clustering technique. *Egyptian Informatics Journal,* 21(4), pp. 201-208.

Shi, Y., Suo, K., Kemp, S. & Hodge, J., 2020. *A Task Scheduling Approach for Cloud Resource Management.* s.l., IEEE, pp. 131-136.

Shojafar, M. et al., 2016. *TETS: a genetic-based scheduler in cloud computing to decrease energy and makespan.* s.l., Springer, pp. 103-115.

Siwani Sharma, D. & Tyagi, S., 2017. COST-BASED TASK SCHEDULING IN CLOUD COMPUTING. *International Research Journal of Engineering and Technology (IRJET),* 4(7), pp. 694-699.

Su, S. et al., 2013. Cost-efficient task scheduling for executing large programs in the cloud. *Parallel Computing,* 39(4-5), pp. 177-188.

Thaman, J. & Singh, M., 2016. Current perspective in task scheduling techniques in cloud computing: a review. *International Journal in Foundations of Computer Science & Technology,* 6(1), pp. 65-85.

Thanka, M. R., Maheswari, P. U. & Edwin, E. B., 2019. A hybrid algorithm for efficient task scheduling in cloud computing environment. *International Journal of Reasoning-based Intelligent Systems,* 11(2), pp. 134-140.

Thiyeb, I. A. & Alhomdy, S. A., 2020. HAMM: A Hybrid Algorithm of Min-Min and Max-Min Task Scheduling Algorithms in Cloud Computing. *International Journal of Recent Technology and Engineering (IJRTE),* 9(4), pp. 209-218.

Tong, Z. et al., 2020. QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment. *Neural Computing & Applications,* 32(10), pp. 5553-5570.

Tran, M.-N. & Kim, Y., 2021. *A Cloud QoS-driven Scheduler based on Deep Reinforcement Learning.* s.l., IEEE, pp. 1823-1825.

Tuli, S. et al., 2022. HUNTER: AI based holistic resource management for sustainable cloud computing. *Journal of Systems and Software,* Volume 184, p. 111124.

uz Zaman, S. K., Tahir Maqsood, M. A. & Bilal, K., 2019. A load balanced task scheduling heuristic for large-

scale computing systems. *Comput. Syst. Sci. Eng,* 34(2), pp. 79-90.
Venu, G. & Vijayanand, K., 2020. Task Scheduling in Cloud Computing: A Survey. *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* 8(V), pp. 2258-2266.
Verma, A. & Kaushal, S., 2015. Cost minimized PSO based workflow scheduling plan for cloud computing. *International Journal of Information Technology and Computer Science,* 7(8), pp. 37-43.
Wikipedia, n.d. *Cloud Computing.* [Online] Available at: https://en.wikipedia.org/wiki/Cloud_computing#cite_note-71 [Accessed 2 10 2021].
Wu, X. et al., 2013. A task scheduling algorithm based on QoS-driven in cloud computing. *Procedia Computer Science,* Volume 17, pp. 1162-1169.
Xue, S. et al., 2017. QET: a QoS-based energy-aware task scheduling method in cloud environment. *Cluster Computing,* 20(4), pp. 3199-3212.
Yadav, R. et al., 2018. Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. *IEEE Access,* Volume 6, pp. 55923-55936.
Youssef, A. E., 2012. Exploring cloud computing services and applications. *Journal of Emerging Trends in Computing and Information Sciences,* 3(6), pp. 838-847.
Yuan, H. et al., 2016. TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds. *IEEE transactions on cybernetics,* 47(11), pp. 3658-3668.
Zhang, G.-h. & Abnoosian, K., 2020. Scheduling mechanisms in the cloud environment: a methodological analysis. *Kybernetes: The International Journal of Systems & Cybernetics,* 49(12), pp. 2977-2992.
Zhang, H., Wu, Y. & Sun, Z., 2021. EHEFT-R: multi-objective task scheduling scheme in cloud computing. *Complex & Intelligent Systems,* pp. 1-8.
Zhang, Q., Cheng, L. & Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications,* 1(1), pp. 7-18.
Zhang, Y. & Xu, B., 2015. Task scheduling algorithm based-on QoS constrains in cloud computing. *International Journal of Grid and Distributed Computing,* 8(6), pp. 269-280.
Zhao, G., 2014. Cost-aware scheduling algorithm based on PSO in cloud computing environment. *International Journal of Grid and Distributed Computing,* 7(1), pp. 33-42.
Zhao, H. et al., 2019. *Energy-efficient task scheduling for heterogeneous cloud computing systems.* s.l., IEEE, pp. 952-959.
Zheng, J. & Wang, Y., 2021. A Hybrid Multi-Objective Bat Algorithm for Solving Cloud Computing Resource Scheduling Problems. *Sustainability,* 13(14), p. 7933.
Zhou, J. & Dong, S., 2018. Hybrid glowworm swarm optimization for task scheduling in the cloud environment. *Engineering Optimization,* 50(6), pp. 949-964.
Zohrati, L., Abadeh, M. N. & Kazemi, E., 2018. Flexible approach to schedule tasks in cloud-computing environments. *IET Software,* 12(6), pp. 474-479.
Zong, Z., 2020. *An Improvement of Task Scheduling Algorithms for Green Cloud Computing.* s.l., IEEE, pp. 654-657.