IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 October 2023, accepted 29 October 2023, date of publication 6 November 2023, date of current version 14 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330434

RESEARCH ARTICLE

Energy Efficient Resource Allocation in Cloud Environment Using Metaheuristic Algorithm

SAURABH SINGHAL[1], NAKUL GUPTA[2], PARVEEN BERWAL[3], QUADRI NOORULHASAN NAVEED[4], AYODELE LASISI[4], AND ANTENEH WOGASSO WODAJO[5]

[1]Department of Computer Engineering and Applications, GLA University, Mathura, Uttar Pradesh 281406, India
[2]Department of Civil Engineering, GLA University, Mathura, Uttar Pradesh 281406, India
[3]Department of Civil Engineering, Galgotias College of Engineering and Technology, Greater Noida, Uttar Pradesh 201310, India
[4]Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[5]Department of Automotive Engineering, College of Engineering and Technology, Dilla University, Dilla 6220, Ethiopia

Corresponding authors: Anteneh Wogasso Wodajo (wogasso.anteneh@gmail.com) and Nakul Gupta (nakul030588@gmail.com)

This work was supported by the Deanship of Scientific Research at King Khalid University under Grant RGP2/466/44.

ABSTRACT Utility-based computing popularly known as ''cloud computing'' offers several computing services to the users. Due to the proliferation in the users of cloud computing, there is an unprecedented increase in the demand for computation resources to execute cloud services. Thus, there is a requirement to investigate currently available resources like virtual machines, CPU, RAM, and storage to allocate cloud services. The allocation and QoS of cloud services are highly dependent on allocation schemes. The optimized solutions allocate resources to submitted jobs to reduce the overall cost to the end-users/service provider without degrading the performance of virtual machines. The allocation techniques also consider the harvesting of energy consumption required for running the cloud services. In this paper, we have utilized a Rock Hyrax-based optimization technique to allocate resources to the submitted jobs with reduced energy consumption. The proposed Rock Hyrax algorithm has been simulated on the CloudSim simulator for various scenarios. The performance of the proposed algorithm has been measured over various Quality of Service (QoS) parameters such as makespan, energy efficiency, response time, throughput, and cost. The gathered results validate the proposed algorithm that improves the QoS parameters by 3%-8% as compared to algorithms when both jobs and resources are considered to be dynamic in nature.

INDEX TERMS Cloud computing, rock hyrax optimization, resource allocation, cost, energy efficiency.

I. INTRODUCTION

In data centers, the cloud services are installed on various virtual machines that execute over dedicated physical machines (high-end servers). Virtual machines offer several advantages to end-users, like mobility, agility, scalability and elasticity. A virtual machine provides an execution environment for cloud services by virtualizing physical machine resources such as CPU, RAM and storage to execute the jobs of the users [1], [2]. One of the major issues in this environment is to provide services without disruption to the end-users which are dynamically increasing and decreasing. It leads to an increase or decrease in the running instances of virtual machines of the dedicated cloud service. As submitted jobs require various resources such as I/O, memory and CPU, the resource allocation techniques ensure the distribution of virtual machines over the physical machine as per the requirements [3]. There are two technical constraints to provide elasticity in the cloud computing environment. Firstly, the resources of the physical machines are confined [4]. Secondly, to execute jobs in the cloud, priorities ought to be in congruity with the increased demand for the available resources.

To deal with the above-mentioned issues, the data centers implement several allocation techniques. The resource allocation schemes may be static or dynamic [5]. In static allocation, resources are allocated before they move to execution. In dynamic allocation, the essential idea is to allocate resources at the time of job execution. In dynamic allocation,

The associate editor coordinating the review of this manuscript and approving it for publication was Nitin Gupta.

apart from cost estimation like in static allocation, decision making and estimation of system state are also important [6]. The main objective of the allocation techniques is to minimize the waiting time and execution time of a submitted job to minimize allocation cost. Popular allocation schemes like FIFO [2] and Round-Robin [7] are implemented in the data centers. However, these techniques are not able to allocate resources (virtual machines) efficiently regardless of task priority [8]. The submitted job by any end-user needs to wait in the waiting queue before the resources required by it are allocated. These submitted jobs are priority-free jobs, i.e., no priority is assigned to any jobs [9]. Traditional methods for resource allocation use uncertain and inaccurate optimization techniques that are very time-consuming and are regularly caught in local maxima [10].

As both the jobs and resources are heterogeneous and dynamic in nature, the current methods for allocating resources require an advanced study of parameters. For example, the end-user who has submitted the job may request from the service provider a large number of resources to run services as per the service level agreement (SLA) and required QoS [3]. However, because the resources are diverse and dispersed in the cloud, scheduling and allocation become hard to manage. Thus, scheduling has to make a schedule plan that is a tradeoff between QoS and cost. This trade-off between QoS and the cost associated with allocating resources is a multi-objective problem [11]. Also, the resource allocation techniques have to focus on multi-objective functions to meet the needs of both end-users and service providers. Therefore, to achieve better resource efficiency and utilization, the exploration and development of new allocation algorithm are required. The growth of meta-heuristic algorithms has seen exceptional growth over the last couple of decades. Scientists have been motivated to use the meta-heuristic algorithm to solve NP-hard problems. The advantage of using such algorithms is that they can find the optimized solution in less computational effort and iteration than simple heuristic algorithms. The characteristics of the meta-heuristic algorithm include simplicity, adaptability, source-free solution, and the ability to escape getting trapped in local optima.

Several authors [12], [13], [14] have attempted to address the problem of resource allocation in the data centers. These solutions to allocate resources have considered various QoS parameters such as makespan, energy efficiency, response time, throughput, and cost. Since the service providers are bound to the SLA of users for the requested resources and QoS in the cloud, it becomes essential for them to examine multiple QoS parameters to allocate resources. Several multi-objective methods like Ant Colony Optimization [15], Particle Swarm Optimization [16], Artificial Bee Colony [17] and Bacterial Foraging Optimization [18] etc. are available in the cloud environment [19]. The authors in [20] considered multi-objective functions on cost and makespan time. A Cuckoo-based algorithm that considered Cost and execution time was given by the authors in [21]. To minimize response time and maximize profit, a PSO-based algorithm was given by authors in [22].

In this paper, we have proposed a novel Rock Hyrax meta-heuristic based resource allocation algorithm that minimizes the cost of resource allocation to end-users and the energy consumption to service providers. The paper uses a multi-objective function for allocating resources on cost and energy in a heterogeneous and dynamic cloud environment. The main idea behind the algorithm is to avoid getting trapped in the problem of local maxima. This is achieved by exploiting and exploring all the possible heuristic solutions for allocating resources dynamically in the cloud environment. The QoS parameters considered in this paper for resource allocation are cost, energy efficiency, throughput, deadline, and makespan time.

Virtual machines available in the cloud environment are different from each other based on the processing power and cost of using them. The jobs submitted by end-users may likewise be also different and may require different resources. Additionally, for executing a job on any resource, time for preparing the resource is also required. The paper focuses on the order of job execution and allocation of resources to the jobs. Improving resource efficiency reduces job waiting time in a queue and lowers allocation costs.

A. OBJECTIVE

The major objectives of the paper are as follows:

1. The submitted job must be executed on allotted virtual machines within the deadline.
2. The average cost of allocation to the user should be minimum.
3. Efficiency of time and cost of allocating jobs is increased

B. CONTRIBUTION

The contributions of the paper are:

1. Proposal of a nature-inspired meta-heuristic scheduling algorithm for the dynamic and heterogeneous cloud environment.
2. To tackle multi-objective optimization problems, such as minimizing makespan and energy consumption.
3. To allocate jobs resources by minimizing the idle time so as to minimize energy consumption.
4. Using Rock Hyrax optimization algorithm to achieve optimum solutions

The rest of the paper is structured as follows: A brief literature survey of various algorithms of resource allocation presented in the state-of-the-art is presented in Section II. The objectives of the proposed work and the problem definition, input, output, and constraints are presented in Section III. The proposed Rock Hyrax algorithm for resource allocation is also described in section III. The result analysis of the proposed algorithm with the algorithm present in the literature is in Section IV Simulation and analytical results are also discussed in section IV. Finally, the conclusion and the future work are described in section V.

II. RELATED WORK

Literature shows that the issue of resource allocation has gotten the attention of many researchers as various solutions have been proposed in the past. Some of the prevalent algorithms related to resource allocation are discussed in this section. To optimize the resources in the cloud environment, resource allocation is one of the key research issues among researchers [23]. To address resource management problems, various surveys in the past have been presented by various researchers like scheduling [24], provisioning [25], and allocation [26], [27]. Also, to manage resources effectively, evolutionary approaches and genetic algorithms are commonly used by researchers to manage resources in the cloud environment. A categorical characterization of different allocation algorithms as presented in the literature is described in Figure 1.

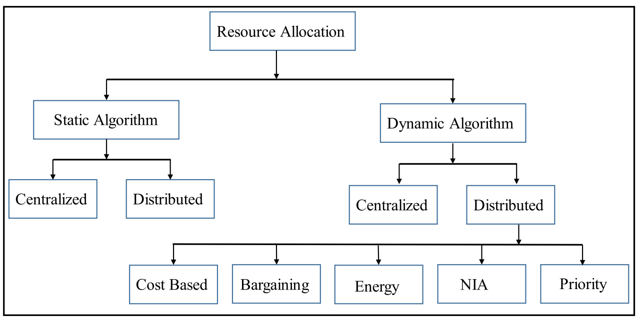


FIGURE 1. Hierarchical taxonomy of allocation algorithms.

The authors in [28], [29], and [30] share the information about resources with cloud providers and end-users for a minimal expense to meet the performance requirement. For resource accounting, the authors in [31], and [32] suggested two different alternatives: one based on usage where each user has a specific number of time units to connect to CPU usage. The other is the pre-allocation capacity of resources. In [33], the authors have used fuzzy systems and standard NSGA-II algorithms for task scheduling in the distributed computing environment. The authors introduced multi-objective functions and aimed to minimize cost and time for implementation while increasing resource productivity. The authors in [34], discuss a resource allocation algorithm using a general heuristic for a workflow application. The main objective of the model is to coordinate the workflow applications and responsibilities allotted to the service. The authors in [35], discuss a model using the Hybrid PSO (HyPSO) for assigning tasks in a distributed environment. The model is used to satisfy the user requirement and to increase productivity by balancing the load on resources. A dynamic model for allocating resources using a dynamic pricing model to maximize the advantage of service providers while considering user demand is proposed in [36].

The concept of energy consumption is discussed by the authors in [44], and [45] in various computing services. With the increase in data centers, the problem of energy consumption has become a major concern. It is difficult to estimate and optimize the energy requirement in a heterogeneous cloud environment. To address the issue of VM allocation by the service provider to physical machines, the authors in [46] propose a VM allocation approach based on auction-based and negotiation-based that reduces energy consumption. The approach discussed considered migration cost. The authors in [47], for allocating resources to meet the demand of cloud users, used Spider Monkey Optimization (SMO) to minimize various QoS parameters.

In cloud computing, the cost of utilizing the resources is an important issue. The cloud user wants the service to be charged at a minimum price and as per the definition of cloud computing, the services must be offered economically [48]. A market-driven auction resource allocation model on-demand based preferential is proposed in [49]. The payment strategy is based on the service preferences of the user. An auction method that uses a game theory model to determine the winner of the auction is proposed by authors in [50]. If adequate information is not available, then the game is repeated. To allocate VMs to user applications, an allocation algorithm was developed by [51]. The problem is solved using a polynomial-time heuristic as it is represented as a resource optimization problem.

The authors in [52] discussed a new algorithm based on ACO to allocate resources in the IaaS cloud. The algorithm initially forecasts the capacity of available resources and then, based on parameters like time and cost procures computing nodes on which tasks would be allocated. To improve responsiveness to customer demand, the authors in [20], proposed an algorithm Spacing Multi-Objective Antlion algorithm (S-MOAL) that minimizes cost and makespan time of VMs. The authors in [21], proposed a resource allocation algorithm for the scenario when resources are insufficient and inappropriate for fulfilling the demand of users. The task submitted by users follow a strict deadline. They proposed an algorithm based on Cuckoo Driven PSO to ensure QoS constraints and profit of service provider.

To utilize idle resources, the authors in [53] proposed an allocation mechanism based on a double combinatorial auction motivated by the methods of microeconomics like flexibility. To make decisions on price, the authors used a backpropagation neural network. For allocating resources in the IaaS cloud, an algorithm based on PSO as Position Balanced Parallel Particle Swarm Optimization (PBPPSO) algorithm is discussed by authors in [22]. The algorithm discovers resources for a group of jobs optimizing cost and makespan time. Table 1 illustrates a parameterized analysis of different meta-heuristic algorithms presented in the past by various researchers for allocating resources in the cloud.

Based on the literature, to efficiently manage resources in the cloud, it is essential to have a comprehensive understanding of resource utilization and optimization strategies.

TABLE 1. Summary of reviewed papers for job allocation algorithms.

Ref	Algorithm	Problem Resolved	Pros	Limitations	QoS parameters
[15]	ACO	Dynamic Resource allocation	Considers network overhead	Workflow are not considered	Cost and execution time
[20]	Ant Lion	Response of customer demand	Multiobjective resource allocation	Workflow are not considered	Makespan, Cost and Energy
[21]	Cuckoo Driven PSO	Optimal resource allocation	Improved performance for large problem size	Only IaaS cloud considered	Cost and Execution Time
[22]	Position based PSO	Optimal allocation	Improved performance	resources are allocated based on learning	response time and profit
[49]	Demand based allocation	Allocation on payment	Improved performance	Priority based allocation	Cost
[52]	ACO	Dynamic resource allocation	Reduce response time	Based on Grid Environment	Time and Cost
[54]	Economic Resource Allocation	Dynamic Resource allocation	predictable, heuristic, and economic	high overhead and complex	Cost
[55]	Grasshopper Optimization Algorithm	Optimized resource allocation	Reduces total cost of messages	Results not elaborative	Cost

To address the environmental impact of cloud computing, it is essential to design algorithms that optimize resource utilization while minimizing energy consumption. Innovative solutions are needed to balance application performance with energy conservation objectives. Dynamic resource allocation strategies that can adapt in real-time to fluctuating workloads and resource demands are growing. A resource allocation that includes energy efficiency, dynamic allocation, and standardized evaluation is required for efficient, secure, and sustainable cloud operations.

The paper presents a model for job allocation in cloud environment over a virtual machine. The model reduces the cost of allocation in a multi-user cloud environment, where the requests are to be executed over a fixed number of virtual machines. We propose the Rock Hyrax Optimization algorithm (RHO) as a solution to the problem and use the CloudSim simulator to simulate the proposed algorithm. We have evaluated and compared the performance of the proposed resource allocation algorithm with metaheuristic algorithms like Ant Colony Optimization [15], Particle Swarm Optimization [16], Artificial Bee Colony [17] and Bacterial Foraging Optimization [18]. The proposed resource allocation algorithm outperformed these metaheuristic algorithms in terms of efficiency and effectiveness.

III. PROPOSED WORK

Optimizing resource allocation in cloud computing is crucial for conserving energy in a data-driven world. Effective resource management is crucial, as cloud data centers are large energy consumers. Dynamic resource allocation techniques, which adjust resource provisioning in real-time based on workload fluctuations, are essential to reduce overprovisioning and provide appropriate resources when needed. Thus, in a cloud environment, to manage resources efficiently while reducing the energy consumption an efficient resource allocation algorithm is required.

In the cloud environment, the service provider has a large pool of virtualized distributed resources like virtual machines and needs to allocate all submitted jobs to different virtual machines. A service provider provides services to many cloud users on a pay-as-you-go basis. Each user individually or in a group submits the job to the cloud environment with its resource requirements, the expected deadline of the job, and other information that is required for the successful execution of the job. The user needs to pay the service provider for the time the resources will be executing their job. In the same manner, different users present at different locations will submit their jobs along with execution details to the cloud environment. The broker monitors state of jobs. The service provider will collect all the jobs and then schedule them with the help of the scheduler. Once the schedule is ready, it is passed to the allocator. The allocator at a particular time as mentioned in the schedule will allocate the suitable resources to the jobs for their execution. The detailed description of the entire process or resource allocation is depicted in Figure 2.

A. PROBLEM FORMULATION

In this section, the job allocation problem is represented by linear programming where the jobs (n) submitted by users are allocated to virtual machines (m). In this work, we have assumed that every individual job is allocated to a single VM; each VM will execute a single job at a given time i.e. one-to-one mapping between resources and jobs is considered. Furthermore, for better utilization of resources, the number of jobs is considered to be greater or equal to resources, i.e. $n \geq m$. $Cost_{ij}$ is the cost of executing the job when Job_i is allocated to $resource_j$.

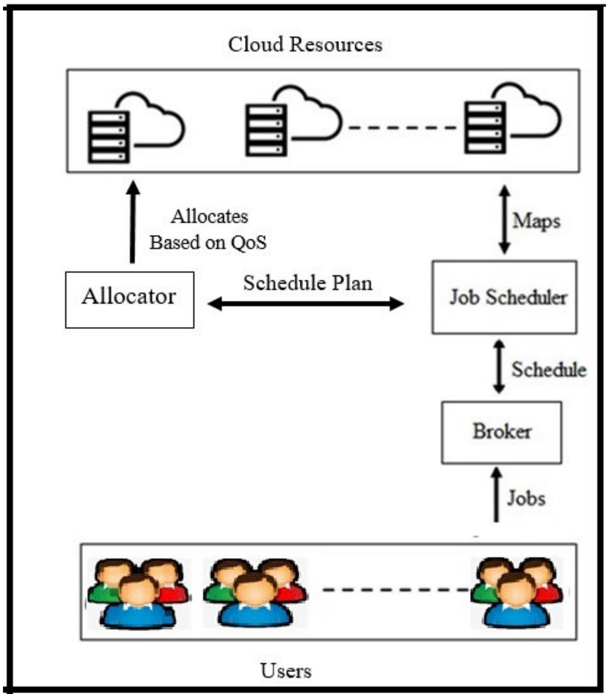


FIGURE 2. Resource allocation process.

The mapping matrix of a job request to a resource group can be represented as:

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix} \tag{1}$$

where C_{ij} is the base price of resource for executing job 'J' Thus, $Cost_{ij}$ mathematically is the product of Job_i is allocated to $resource_j$ and can be expressed as

$$\mathrm{Cost}_{ij} = \mathrm{Job}_i * \mathrm{Resource}_j \tag{2}$$

subjected to i, j > 1.

Therefore, if job_i is mapped to $resource_j$, then the mapping can be represented as:

$$C_{ij} = \begin{cases} C_{ij}, & \text{if } \mathrm{Job}_i \text{is allcoated to } resource_j \\ 0, & else \end{cases} \tag{3}$$

The proposed work aims to minimize the cost of allocation to the users and the energy consumption by the resources using multi-objective optimization. Thus, the objective function for the proposed work can be mathematically expressed as:

$$\min [cost] = \sum_{i=1}^{n} \sum_{j=1}^{m} job_i * resource_j \tag{4}$$

subjected to, $\sum_{i=1}^{n} Job_i = 1$ for i = 1,2,3…n and $\sum_{j=1}^{m} Resource_j = 1$ for j = 1,2,3…m such that Job_i, $Resource_j > 0$, i, j ∈ N = 1,2,….n.

such that, Job_{ij}, $Resource_{ij} > 0$. i, j ∈ N = 1,2,….n

The constraints must satisfy the relation that all the jobs are mapped to available free resources.

$$\min \left[Energy_{total}\right] = \sum_{i=1}^{m} \int_{S_{Time}}^{F_{time}} E_i(T, U) \tag{5}$$

where T is the specific time, U is the utilization factor, $Energy_{total}$ is the total energy used by physical machines at data centers, S_{time} is the starting time, F_{time} is the end time and E_i is the total amount of energy utilized by resources between S_{time} and F_{time}. Many assumptions have been considered while carrying out this study. Many assumptions have been considered while carrying out this study.

These assumptions are as follows:

1. Virtual machines and resources are the same entity.
2. Jobs are considered to be independent.
3. Environment for simulation is heterogeneous and dynamic.
4. Execution of all submitted jobs is compulsory
5. Every job will be executed only by one virtual machine.
6. Each virtual machine in the environment has a different processing speed and allocation cost.

Thus, the resource allocation algorithm is converted into the solution of a mathematical model for multi-objective functions. This model is NP-hard in nature as the solution is not unique but versatile. These solutions cannot be compared, however, can be reached using a multi-objective evolutionary algorithm.

B. PROPOSED RESOURCE ALLOCATION ALGORITHM

Rock hyraxes are small-sized mammals and are vegetarian in nature [56]. Their foraging behaviour mimics the Divide and Conquer technique and is usually in groups of 80-100 during mid-morning and evening. One member of the group acts as a sentinel and monitors the surrounding for other members from predators [57]. Food searching is the responsibility of male Hyrax who inform other members once foraging is successful. To secure the group, searching for food is restricted to a limit. For communication, the Hyrax produces different sounds where each sound has a different meaning. Rock Hyrax optimization strategy is used for optimizing the allocation of jobs to VMs. The process flow for the proposed strategy is depicted in Figure 3.

The population of Rock Hyrax in the problem space is initialized as RHi in the proposed algorithm. Algorithm 1 describes the Rock Hyrax-based resource allocation mechanism. The various data structures used in the proposed algorithm are as:

RHt total count of Rock Hyrax available in the problem space

VMt total count of VMs available in the problem space

Selected VMt is the selected virtual machine for allocating a job and Selected VMt ∈ VMt

Other VMnum is the difference between VMtotal and Selected VMtotal

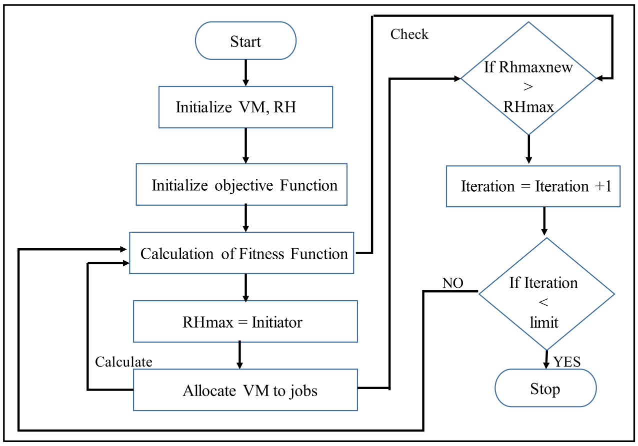


FIGURE 3. Proposed algorithm flowchart.

Ch is the maximum cost of the solution given by the algorithm.

Rock Hyrax-based algorithm for allocating jobs is given in Algorithm 1:

Algorithm 1 Proposed Rock Hyrax Optimization Algorithm for Resource Allocation

Input: $Prob_{size}$, RH_t, VM_t, $PrivilegedVM_{total}$, Privileged RH_{total}, $OtherVM_{total}$
Result: RH_{best} Pop ← IntializePop(RH_{total}, $Problem_{size}$)
$RH_{maxcost}$ ← Cost(Ch)
while StopConditon() do
    EvaluatePop(Pop)
    RH_{best} ← GetBestSolution(Pop)
    VM_{max} ← SelectBestVM(Population, VMt)
    foreach VMi ∈ VMmax do
        $SelectedRH_t \leftarrow \phi$
        if i < SelectedVMt then
            $SelectedRH_t$ ← RH_t
        end
        else
            $SelectedRH_t$ ← $OtherRH_t$
        end
    end
    $RemainingRH_t$ ← (RH_t - VM_t)
end
Return RH_{best}

IV. EMPIRICAL EVALUATION

For resource allocation in the cloud environment, the proposed Rock Hyrax is a nature-inspired algorithm motivated by meta-heuristic algorithms and is represented as a min-objective problem for cost and energy. The population of Rock Hyrax is input to the algorithm and the algorithm finds the fitness function at every iteration while allocating the jobs of virtual machines. The Hyrax that has the best fitness value is chosen as Universal Rock Hyrax and is responsible for foraging.

A. PERFORMANCE METRICS

In cloud environment, to address the allocation problem, various researchers like [12], [13], and [14] have proposed solutions. These solutions, for allocation, consider only a single QoS parameter. Since in the cloud environment the users pay for the resources it uses, it becomes essential for allocation algorithms to examine multiple QoS parameters. The QoS parameters considered in this paper for resources allocation are:

Makespan: It is the total time required by a Job J_i on resource R_j to completely get executed.

where,

$$MS = \max(ET_{ij}) \tag{6}$$

ET_{ij} is the execution time of the job Ji on resource R_j.

Cost: It is the sum that end users must pay for using the resources to carry out tasks in the cloud environment. It is a source of revenue for service providers while costing customers [58]. The cost can be calculated as:

$$Cost_{total} = \sum_{i=1}^{m} (Cost_i * Time_i) \tag{7}$$

where,

m is the total number of resources available

$Cost_{total}$ is total cost of allocating all the submitted jobs

$Cost_i$ is the cost of allocating $resource_i$ to Job_i

$Time_i$ is the time of utilization of $resource_i$ to Job_i

Energy: It is the amount of power required by resources for executing jobs in cloud computing [59]. It is the electricity required by the data centers to operate physical machines. The energy consumption of $resource_i$ at specific time T with utilization factor U is given by:

$$Energy_{total} = \sum_{i=1}^{m} \int_{S_{Time}}^{F_{time}} E_i(T, U) \tag{8}$$

where,

$Energy_{total}$ is the total energy used by physical machines at data centers

S_{time} is the starting time of resource utilization

F_{time} is the end time of resource utilization

E_i is the amount of energy consumed by $resource_i$

Throughput: The number of tasks that are successfully executed in a given time in the cloud environment.where,

$$\text{Throughput} = \sum_{i=1}^{n} Exec_{Time} \tag{9}$$

$Exec_{Time}$ is the execution time of jobi

Response time: It is the time for a task from its submission to the time when the resources are allocated to it or when a task starts its execution after waiting in the waiting queue [60].

where,

$$RT = \sum_{i=1} (Sub_{Time} + Start_{Time}) \tag{10}$$

Sub_{time} is the submission time of the task

$Start_{time}$ is the time when the execution of the task starts.

B. EXPERIMENTAL SETUP

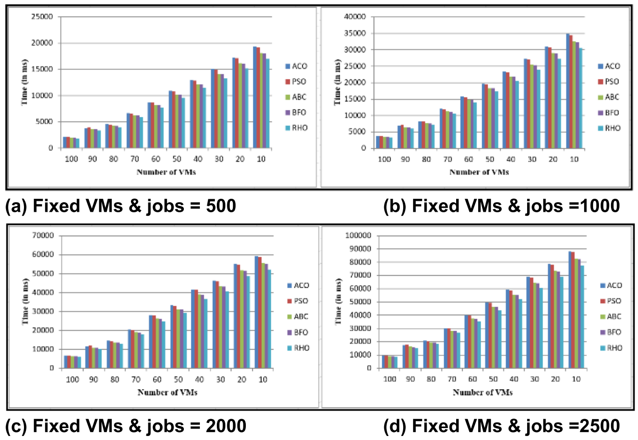
The proposed job allocation algorithm is implemented on CloudSim 3.0.3 Windows 7 desktop edition simulator. CloudSim simulates the cloud environment by creating cloudlets as jobs, data centers and virtual machines. To simulate the proposed algorithm in CloudSim, eight data centers have been created. The experimental results are achieved after implementing various algorithms like ACO, PSO, ABC, and BFO on the CloudSim environment when both jobs and virtual machines are kept dynamic. The performance of the algorithm is measured on the following QoS parameters: Makespan time, response time, cost, energy efficiency and throughput. The experimental results are obtained after running different algorithms by varying both jobs and resources in the simulated environment over two different scenarios. In Scenario-I VMs are varied from 10 to 100 while jobs are fixed and in scenario-II jobs are varied while keeping VMs fixed. The details of experimental setup for both scenarios are shown in Table 2. The length of jobs was varied by considering different length of jobs to represent the cloud environment.

TABLE 2. Experimental setup for scenario I & II.

Entity	Variable	Scenario I	Scenario II
User	Cloudlets	10-100	10-100
Cloudlets	Length	500-15000	250-10000
Host	Hosts	8	4
	RAM	16 GB	16 GB
	Storage	1 TB	1 TB
	Bandwidth	512	512
VM	VMs	8	10-100
	RAM	4GB	4GB
	OS	Windows	Windows
	Policy	Time Sharing	Time Sharing
	CPUs	4	4
Data Centers	Data Centers	8	8

C. RESULT ANALYSIS FOR SCENARIO I

In order to execute the algorithms under Scenario-I, where the number of virtual machines remains constant while the number of jobs varies from 10 to 100, the experimental parameter settings of CloudSim are illustrated in Table 2. The table provides a detailed overview of the parameters that were set for the experiments, including the number of data centers, hosts, virtual machines, and cloudlets. These parameters were chosen to ensure that the experiments were conducted under controlled conditions and to enable a fair comparison of the different allocation algorithms that were tested. The table also lists the values that were assigned to various parameters such as the VM scheduling policy, the time zone, and the utilization threshold. This detailed information is essential for understanding the experimental setup and for replicating the experiments in future studies. Overall, the experimental parameter settings of CloudSim in Scenario-1 were carefully selected to ensure that the results obtained were reliable and could be used to inform future research in the field of cloud resource allocation.



(a) Fixed VMs & jobs = 500 (b) Fixed VMs & jobs =1000

(c) Fixed VMs & jobs = 2000 (d) Fixed VMs & jobs =2500

FIGURE 4. Makespan time for scenario I when (a) jobs is 500, (b) VM is 1000, (c) VM is 2000 and (d) jobs is 2500.

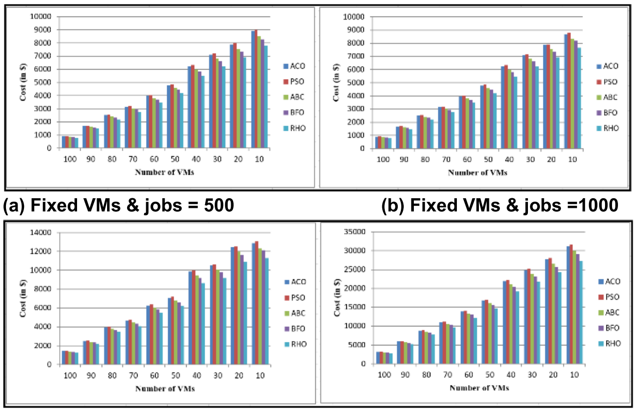
Figure 4 presents a detailed analysis of makepan time for different allocation algorithms in Scenario I, where the number of virtual machines is varied from 10 to 100 and the number of jobs is constant at 500, 1000, 2000 and 2500. The results show that the performance of the algorithms is equivalent for a smaller number of jobs, but as the number of jobs increases, the proposed algorithm outperforms the others. The proposed algorithm achieves better results by avoiding local minima that can negatively impact the performance of other algorithms and by selecting the best fitness function calculated during iterations.

(a) Fixed VMs & jobs = 500 (b) Fixed VMs & jobs =1000

(c) Fixed VMs & jobs = 2000 (d) Fixed VMs & jobs =2000

FIGURE 5. Response time for scenario I when (a) jobs is 500, (b) VM is 1000, (c) VM is 2000 and (d) jobs is 2500.

Response time for the proposed allocation algorithm for scenario I is shown in Figure 5. The number of virtual machines varies from 10 to 100 and the number of jobs is constant, is compared to existing algorithms. The results show that the algorithm effectively minimizes response time by continuously searching for free resources to allocate to jobs. This reduces the response time of submitted jobs, enhancing the quality of service for end-users. As the number of virtual machines increases, the algorithm can find more free resources to allocate, outperforming existing algorithms in the literature. The comparison highlights the advantages of the proposed algorithm in terms of response time and its ability to improve cloud service performance.

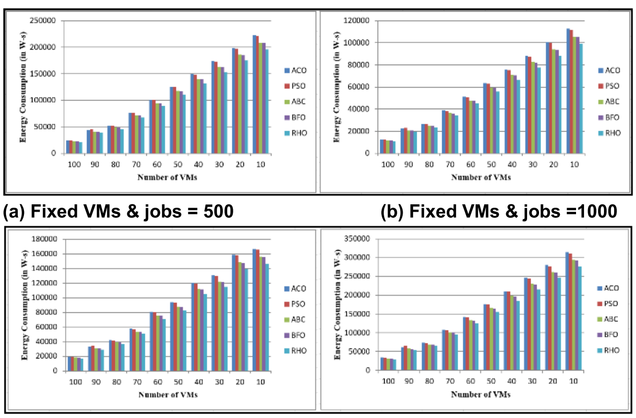


(a) Fixed VMs & jobs = 500 (b) Fixed VMs & jobs =1000

(c) Fixed VMs & jobs = 2000 (d) Fixed VMs & jobs =2500

FIGURE 6. Cost for scenario I when (a) jobs is 500, (b) VM is 1000, (c) VM is 2000 and (d) jobs is 2500.

Figure 6 provides a detailed analysis of the cost of allocating resources to a job. The proposed algorithm can minimize end-user costs when the number of virtual machines increases while jobs are fixed. However, if limited VMs are available, the cost is comparable to existing literature. This comparison helps make informed decisions about resource allocation and optimizes end-user costs, enhancing the efficiency of the proposed algorithm.

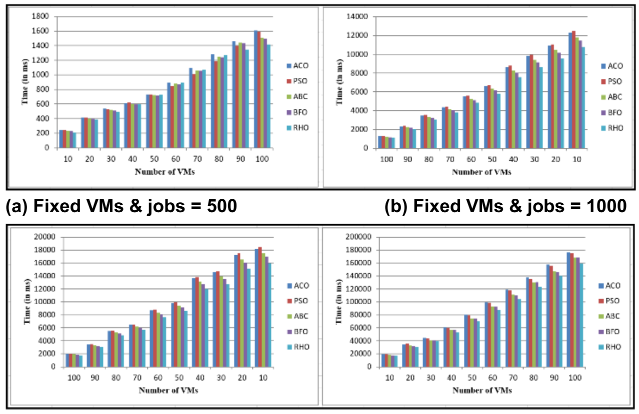


(a) Fixed VMs & jobs = 500 (b) Fixed VMs & jobs =1000

(c) Fixed VMs & jobs = 2000 (d) Fixed VMs & jobs =2500

FIGURE 7. Energy consumption for scenario I when (a) jobs is 500 (b) VM is 1000, (c) VM is 2000 and (d) jobs is 2500.

Figure 7 shows that as the number of virtual machines (VMs) increases while maintaining the physical machine (PM) constant, the energy required for idle tasks also increases. This is due to increased resource demand. However, a proposed algorithm can reduce energy consumption by evenly distributing load across different datacenters, leading to energy-efficient resource allocation. This can benefit data centers and the environment by aiding in the design of sustainable data centers that optimize energy consumption while meeting end-user demands.

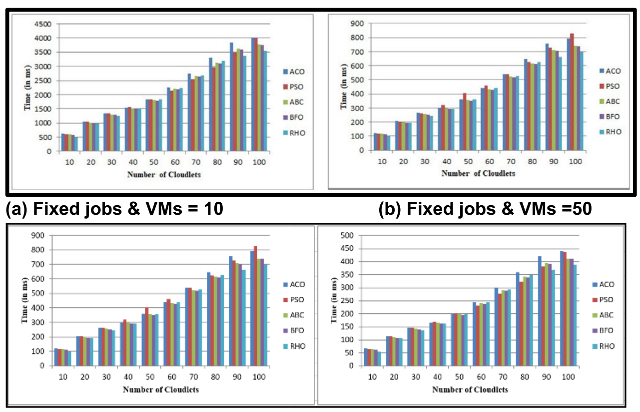


(a) Fixed VMs & jobs = 500 (b) Fixed VMs & jobs = 1000

(c) Fixed VMs & jobs = 2000 (d) Fixed VMs & jobs =2500

FIGURE 8. Throughput for scenario I when (a) jobs is 500, (b) VM is 1000, (c) VM is 2000 and (d) jobs is 2500.

Figure 8 shows a comparison of throughputs for Scenario I, revealing the proposed algorithm as the most efficient. It reduces job duration and response time, resulting in improved throughput values. The algorithm's advantages become more evident as the number of virtual machines increases. When the number of VMs is low, all algorithms show the same level of throughput. However, as the number of VMs increases, the proposed algorithm outperforms other algorithms significantly. This demonstrates the algorithm's potential for effective resource allocation in a cloud computing environment.



(a) Fixed jobs & VMs = 10 (b) Fixed jobs & VMs =50

(c) Fixed jobs & VMs = 75 (d) Fixed jobs & VMs =100

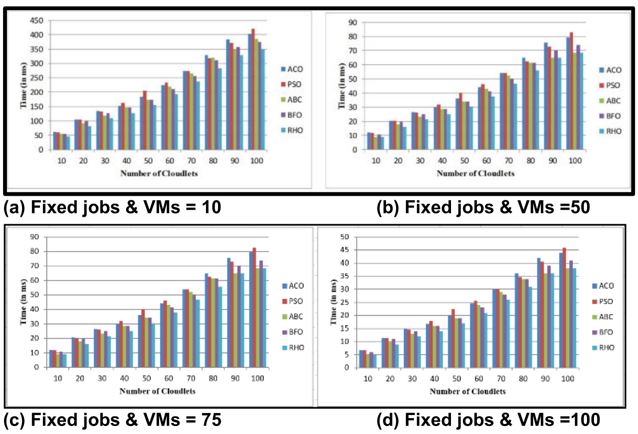
FIGURE 9. Makespan time for scenario II when (a) VM is 10, (b) VM is 50, (c) VM is 75 and (d) VM is 100.

D. RESULT ANALYSIS FOR SCENARIO II

This subsection offers a thorough study of the outcomes of applying scenario II to the suggested algorithm. In this case, the number of virtual machines (VMs) stayed constant, but the number of workloads varied in steps of 10 from 10 to 100.

Table 2 displays the experimental CloudSim parameter settings that were utilized to run the algorithms under scenario II. This table offers a clear and thorough explanation of the experimental parameters, which is crucial for guaranteeing the validity and dependability of the results. The provided data is anticipated to make it easier for other researchers to replicate the experiment and to compare and assess how well various algorithms work in a cloud computing environment.

The proposed algorithm's makepan time is compared under various scenarios, with the number of jobs varying from 10 to 100 and the number of virtual machines (VMs) constant at 10, 50, 75 and 100. The algorithm's performance is comparable to literature algorithms and fixed VMs when jobs are low. However, as jobs increase with a fixed number of VMs, the algorithm outperforms other algorithms by avoiding local maxima problems, resulting in a significant improvement in make-up time. This comparison demonstrates the algorithm's potential for effectively allocating resources in a cloud computing environment, especially when jobs are high and VMs are relatively low.



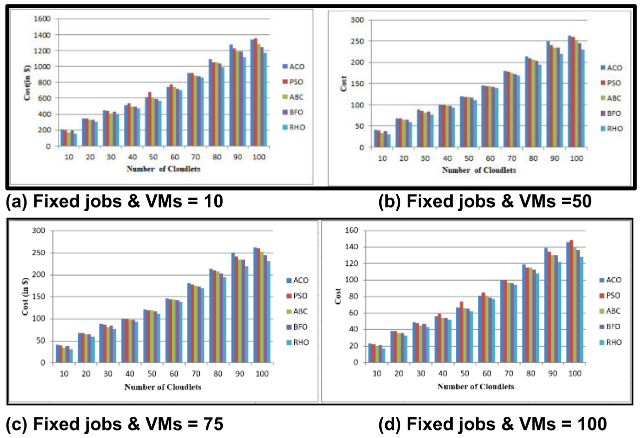
(a) Fixed jobs & VMs = 10 (b) Fixed jobs & VMs =50

(c) Fixed jobs & VMs = 75 (d) Fixed jobs & VMs =100

FIGURE 10. Response time for scenario II when (a) VM is 10, (b) VM is 50, (c) VM is 75 and (d) VM is 100.

The proposed algorithm efficiently allocates resources to jobs with minimal load, reducing job waiting times and improving response time compared to other algorithms. This is particularly effective in a cloud computing environment, especially when the number of jobs is high and the number of virtual machines is low. Figure 10 depicts the performance of response time for Scenario II. As jobs increase, with VMs remain constant, the proposal searches for the resources having minimum load and allocates the resources to the jobs. As a result, the jobs spent less time in the waiting queue improving the response time of the proposed algorithm over others.

Figure 11 shows a comparison of the cost of allocating resources to jobs in a cloud computing environment. As the number of jobs increases with a fixed number of virtual machines (VMs), the allocator has limited options for allocating resources with minimum cost. However, the proposed algorithm efficiently allocates resources with minimum execution costs and reduced waiting times,



(a) Fixed jobs & VMs = 10 (b) Fixed jobs & VMs =50

(c) Fixed jobs & VMs = 75 (d) Fixed jobs & VMs = 100

FIGURE 11. Cost for scenario II when (a) VM is 10, (b) VM is 50, VM is 75 and (d) VM is 100.

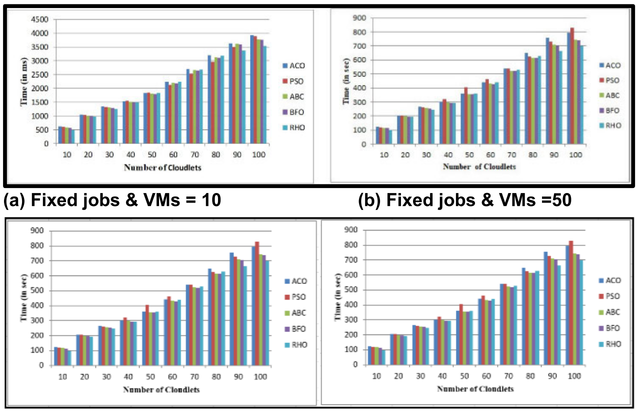
resulting in a significant reduction in end-user costs. This demonstrates the algorithm's potential for effective resource allocation, especially when the number of jobs is high and the number of VMs is low. This information demonstrates the algorithm's effectiveness in minimizing resource allocation costs, crucial for optimal utilization in a cloud computing environment.

(a) Fixed jobs & VMs = 10 (b) Fixed jobs & VMs =50

(c) Fixed jobs & VMs = 75 (d) Fixed jobs & VMs =100

FIGURE 12. Energy consumption for scenario II when (a) VM is 10, (b) VM is 50, (c) VM is 75 and (d) VM is 100.

Figure 12 depicts the comparison of energy consumption of allocating the resource to a job. The proposed algorithm for resource allocation in cloud computing effectively minimizes the quality of service (QoS) parameters, reducing execution and idle time of jobs and servers. This results in a significant reduction in energy required for running data centers. The algorithm's effectiveness in improving energy efficiency in data centers is crucial for sustainable and cost-effective cloud computing services. The results also highlight the potential impact on reducing carbon emissions and overall environmental sustainability. The algorithm's efficiency in identifying available resources and allocating jobs effectively demonstrates its potential in reducing energy consumption in cloud computing environments.



(a) Fixed jobs & VMs = 10
(b) Fixed jobs & VMs =50
(c) Fixed jobs & VMs = 75
(d) Fixed jobs & VMs =100

FIGURE 13. Throughput for scenario II when (a) VM is 10, (b) VM is 50, (c) VM is 75 and (d) VM is 100.

The accuracy of a cloud simulation is significantly influenced by the fidelity of the model used. Due to the complexity of cloud infrastructures, creating an exact model is challenging. The results may not accurately reflect real-world performance due to the model's inability to fully capture real-world cloud activity. The experiments used a synthetic dataset, and jobs were autonomous and undivided tasks.

V. DISCUSSION

This paper proposes an algorithm for allocating jobs on virtual machines using a nature-inspired meta-heuristic algorithm. The proposed RHO algorithm for resource allocation addresses problems faced by cloud service providers, which include energy consumption and cost. The proposed algorithm minimizes overall makespan time and energy efficiency too because it allocates resources to jobs based on availability and load. For the performance evaluation of the proposed algorithms, consider two scenarios. For the first scenario, the jobs are varied in an interval of 10 from 10 to 100 by keeping VM constant. Whereas, in the second scenario, the jobs are kept constant while varying the VM in a gap of 10 from 10 to 100. The two scenarios ensure that performance is measured on both jobs and resources, which are dynamic. Performance comparison through QoS reveals that the proposed algorithm manages geographically distributed resources efficiently by making use of Rock Hyrax optimization. The proposed Rock Hyrax algorithm also addresses the problem of local maxima, which affects the performance of various job allocation algorithms and optimizes energy consumption. The proposed algorithm is compared with other job allocation algorithms proposed in the past and empirically proves that it compares well for both jobs and virtual machines in a static and dynamic environment.

The proposed algorithm has certain advantages over the algorithms present in the literature, as it works on the principle of divide and conquer decreasing the time required to find an optimal mapping. Also, the algorithm avoids local minima, thus providing a better solution. However, if the number of jobs or the number of VMs is less, then the performance of the proposed algorithm remains on par with other algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper, an algorithm for allocating jobs on virtual machines using a nature-inspired based meta-heuristic algorithm that mimics the behavior of Rock Hyrax has been proposed. The proposed RHO algorithm highlights the important problems faced by cloud service providers, including energy consumption and cost. The proposed algorithm minimizes overall makespan time and energy efficiency, as it allocates the job to resources based on availability and current load. For evaluating the performance of the proposed algorithms, two scenarios were used. In the first scenario, the jobs are varied in a gap of 10 from 10 to 100 keeping VM constant. Whereas, in the second scenario, the jobs are kept constant while varying the VM in a gap of 10 from 10 to 100. Performance comparison through QoS reveals that the proposed algorithm manages geographically distributed resources efficiently. The proposed Rock Hyrax algorithm removes the problem of local maxima which affects the performance of various job allocation algorithms and performs energy optimization. The proposed algorithm is compared with other job allocation algorithms proposed in the past and empirically proves that it works well for both jobs and Virtual Machines statically and dynamically.

In the future, we would like to run the algorithm in a real cloud environment. Also, the work can be extended by considering the cost involved in the transportation of jobs and data and the energy required by other components such as memory and hard drives. Further, workflow applications and real datasets can be tested over the proposed work.

CONFLICT OF INTEREST

None

REFERENCES

[1] S. A. Alsaidy, A. D. Abbood, and M. A. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2370–2382, Jun. 2022.

[2] J. Yao and J.-H. He, "Load balancing strategy of cloud computing based on artificial bee algorithm," in *Proc. 8th Int. Conf. Comput. Technol. Inf. Manage. (NCM ICNIT)*, vol. 1, Apr. 2012, pp. 185–189.

[3] S. Zhao, X. Lu, and X. Li, "Quality of service-based particle swarm optimization scheduling in cloud computing," in *Proc. 4th Int. Conf. Comput. Eng. Netw.*, 2015, pp. 235–242.

[4] Q. Zhang, Y. Lin, and Z. Wang, "Cost-effective capacity migration of peer-to-peer social media to clouds," *Peer-to-Peer Netw. Appl.*, vol. 6, no. 3, pp. 247–256, Sep. 2013.

[5] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, S. U. Khan, and A. Zomaya, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, Jul. 2016.

[6] A. Belgacem, "Dynamic resource allocation in cloud computing: Analysis and taxonomies," *Computing*, vol. 104, no. 3, pp. 681–710, Mar. 2022.

[7] U. Ayesta, M. Erausquin, E. Ferreira, and P. Jacko, "Optimal dynamic resource allocation to prevent defaults," *Oper. Res. Lett.*, vol. 44, no. 4, pp. 451–456, Jul. 2016.

[8] A. A. Khan, M. Zakarya, R. Khan, I. U. Rahman, M. Khan, and A. U. R. Khan, ''An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters,'' *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102497.
[9] H. G. Tani and C. El Amrani, ''Cloud computing CPU allocation and scheduling algorithms using CloudSim simulator,'' *Int. J. Electr. Comput. Eng.*, vol. 6, no. 4, p. 1866, Aug. 2016.
[10] A. Khetan, V. Bhushan, and S. C. Gupta, ''Anovel survey on load balancing in cloud computing,'' *J. Eng. Technol.*, vol. 6, no. 3, pp. 1–9, 2013.
[11] A. Mosavi and A. Vaezipour, ''Reactive search optimization; application to multiobjective optimization problems,'' *Appl. Math.*, vol. 3, no. 10, pp. 1572–1582, 2012.
[12] X. Wang, X. Liu, L. Fan, and X. Jia, ''A decentralized virtual machine migration approach of data centers for cloud computing,'' *Math. Problems Eng.*, vol. 2013, pp. 1–10, Jul. 2013.
[13] D. C. Erdil, ''Autonomic cloud resource sharing for intercloud federations,'' *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1700–1708, Sep. 2013.
[14] M. Zukerman, L. Tan, H. Wang, and I. Ouveysi, ''Efficiency-fairness tradeoff in telecommunications networks,'' *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 643–645, Jul. 2005.
[15] P. Gupta, U. Goyal, and V. Verma, ''Cost-aware ant colony optimization for resource allocation in cloud infrastructure,'' *Recent Adv. Comput. Sci. Commun.*, vol. 13, no. 3, pp. 326–335, Aug. 2020.
[16] A.-P. Xiong and C.-X. Xu, ''Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center,'' *Math. Problems Eng.*, vol. 2014, pp. 1–8, Jun. 2014.
[17] A. P. Sheetal and K. Ravindranath, ''Priority based resource allocation and scheduling using artificial bee colony (ABC) optimization for cloud computing systems,'' *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, no. 6, pp. 39–44, 2019.
[18] A. Dhingra, ''Green cloud: Heuristic based BFO technique to optimize resource allocation,'' *Indian J. Sci. Technol.*, vol. 4, no. 5, pp. 685–691, May 2014.
[19] S. Zhang and Y. Zhou, ''Grey wolf optimizer based on Powell local optimization method for clustering analysis,'' *Discrete Dyn. Nature Soc.*, vol. 2015, pp. 1–17, Nov. 2015.
[20] A. Belgacem, K. Beghdad-Bey, H. Nacer, and S. Bouznad, ''Efficient dynamic resource allocation method for cloud computing environment,'' *Cluster Comput.*, vol. 23, no. 4, pp. 2871–2889, Dec. 2020.
[21] A. S. Banu and W. R. Helen, ''Scheduling deadline constrained task in hybrid IaaS cloud using cuckoo driven particle swarm optimization,'' *Indian J. Sci. Technol.*, vol. 8, no. 16, pp. 1–6, Aug. 2015.
[22] R. S. Mohana, ''A position balanced parallel particle swarm optimization method for resource allocation in cloud,'' *Indian J. Sci. Technol.*, vol. 8, no. S3, pp. 182–188, Feb. 2015.
[23] S. S. Manvi and G. K. Shyam, ''Resource management for infrastructure as a service (IaaS) in cloud computing: A survey,'' *J. Netw. Comput. Appl.*, vol. 41, pp. 424–440, May 2014.
[24] S. Singh and I. Chana, ''A survey on resource scheduling in cloud computing: Issues and challenges,'' *J. Grid Comput.*, vol. 14, no. 2, pp. 217–264, Jun. 2016.
[25] S. Singh and I. Chana, ''Cloud resource provisioning: Survey, status and future research directions,'' *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 1005–1069, Dec. 2016.
[26] N. Akhter and M. Othman, ''Energy aware resource allocation of cloud data center: Review and open issues,'' *Cluster Comput.*, vol. 19, no. 3, pp. 1163–1182, Sep. 2016.
[27] A. Beloglazov, J. Abawajy, and R. Buyya, ''Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,'' *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.
[28] S. D. Patil and S. C. Mehrotra, ''Resource allocation and scheduling in the cloud,'' *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 1, no. 1, pp. 47–52, 2012.
[29] R. K. Sharma and N. Sharma, ''A dynamic optimization algorithm for task scheduling in cloud computing with resource utilization,'' *Int. J. Sci. Eng. Technol.*, vol. 2, no. 10, pp. 1062–1068, 2013.
[30] B. M. Lavanya and C. S. Bindu, ''Systematic literature review on resource allocation and resource scheduling in cloud computing,'' *Int. J. Adv. Inf. Technol.*, vol. 6, no. 4, pp. 1–15, Aug. 2016.
[31] N. Asha and G. R. Rao, ''A review on various resource allocation strategies in cloud computing,'' *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 7, pp. 12–28, 2013.
[32] S. Jayanthi, ''Literature review: Dynamic resource allocation mechanism in cloud computing environment,'' in *Proc. Int. Conf. Electron., Commun. Comput. Eng. (ICECCE)*, Nov. 2014, pp. 279–281.
[33] R. Salimi, H. Motameni, and H. Omranpour, ''Task scheduling using NSGA II with fuzzy adaptive operators for computational grids,'' *J. Parallel Distrib. Comput.*, vol. 74, no. 5, pp. 2333–2350, May 2014.
[34] B. Cheng, ''Hierarchical cloud service workflow scheduling optimization schema using heuristic generic algorithm,'' *Przeglad Elektrotechniczny*, vol. 88, no. 2, pp. 92–95, 2012.
[35] K. Krishnasamy, ''Task scheduling algorithm based on hybrid particle swarm optimization in cloud computing environment,'' *J. Theor. Appl. Inf. Technol.*, vol. 55, no. 1, pp. 124–134, 2013.
[36] Z. Qian, X. Wang, X. Liu, X. Xie, and T. Song, ''An approach to dynamically assigning cloud resource considering user demand and benefit of cloud platform,'' *Computing*, vol. 102, no. 8, pp. 1817–1842, Aug. 2020.
[37] R. Eberhart and J. Kennedy, ''A new optimizer using particle swarm theory,'' in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, 1995, pp. 39–43.
[38] G. K. Shyam, S. S. Manvi, and B. Prasad, ''Concurrent and cooperative negotiation of resources in cloud computing: A game theory based approach,'' *Multiagent Grid Syst.*, vol. 14, no. 2, pp. 177–202, Jun. 2018.
[39] Z. Zhou, H. Zhang, X. Yu, and J. Guo, ''Continuous resource allocation in cloud computing,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 319–324.
[40] E. Sherzer and H. Levy, ''Resource allocation in the cloud with unreliable resources,'' *Perform. Eval.*, vol. 137, Feb. 2020, Art. no. 102069.
[41] H. Godhrawala and R. Sridaran, ''A dynamic Stackelberg game based multi-objective approach for effective resource allocation in cloud computing,'' *Int. J. Inf. Technol.*, vol. 15, no. 2, pp. 803–818, Feb. 2023.
[42] A. K. Sangaiah, A. Javadpour, P. Pinto, S. Rezaei, and W. Zhang, ''Enhanced resource allocation in distributed cloud using fuzzy metaheuristics optimization,'' *Comput. Commun.*, vol. 209, pp. 14–25, Sep. 2023.
[43] N. Manikandan, P. Divya, and S. Janani, ''BWFSO: Hybrid blackwidow and fish swarm optimization algorithm for resource allocation and task scheduling in cloud computing,'' *Mater. Today, Proc.*, vol. 62, pp. 4903–4908, Jan. 2022.
[44] H. Chen, Y. Wen, and Y. Wang, ''An energy-efficient method of resource allocation based on request prediction in multiple cloud data centers,'' *Concurrency Comput., Pract. Exp.*, vol. 35, no. 9, Apr. 2023, Art. no. e7636.
[45] W.-B. Sun, J. Xie, X. Yang, L. Wang, and W.-X. Meng, ''Efficient computation offloading and resource allocation scheme for opportunistic access fog-cloud computing networks,'' *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 2, pp. 521–533, Apr. 2023.
[46] W. Wang, Y. Jiang, and W. Wu, ''Multiagent-based resource allocation for energy minimization in cloud computing systems,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 205–220, Feb. 2017.
[47] J. K. Samriya and N. Kumar, ''Spider monkey optimization based energy-efficient resource allocation in cloud environment,'' *Trends Sci.*, vol. 19, no. 1, p. 1710, Jan. 2022.
[48] L. Qian, Z. Luo, Y. Du, and L. Guo, ''Cloud computing: An overview,'' in *Proc. IEEE Int. Conf. Cloud Comput.*, Dec. 2009, pp. 626–631.
[49] N. Kumar and S. Saxena, ''A preference-based resource allocation in cloud computing systems,'' *Proc. Comput. Sci.*, vol. 57, pp. 104–111, Jan. 2015.
[50] A. Nezarat and G. Dastghaibifard, ''Efficient Nash equilibrium resource allocation based on game theory mechanism in cloud computing by using auction,'' *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0138424.
[51] K. Kumar, J. Feng, Y. Nimmagadda, and Y.-H. Lu, ''Resource allocation for real-time tasks using cloud computing,'' in *Proc. 20th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2011, pp. 1–7.
[52] W. X. Hu, J. Zheng, X. Y. Hua, and Y. Q. Yang, ''A computing capability allocation algorithm for cloud computing environment,'' *Appl. Mech. Mater.*, vols. 347–350, pp. 2400–2406, Aug. 2013.
[53] J. Sun, X. Wang, K. Li, C. Wu, M. Huang, and X. Wang, ''An auction and league championship algorithm based resource allocation mechanism for distributed cloud,'' in *Proc. Int. Workshop Adv. Parallel Process. Technol.*, 2013, pp. 334–346.
[54] M. Babaioff, E. Timnat, Y. Mansour, N. Nisan, G. Noti, C. Curino, N. Ganapathy, I. Menache, O. Reingold, and M. Tennenholtz, ''ERA: A framework for economic resource allocation for the cloud,'' in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 635–642.

[55] J. Vahidi and M. Rahmati, "Optimization of resource allocation in cloud computing by grasshopper optimization algorithm," in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Feb. 2019, pp. 839–844.
[56] D. J. Druce, J. S. Brown, J. G. Castley, G. I. H. Kerley, B. P. Kotler, R. Slotow, and M. H. Knight, "Scale-dependent foraging costs: Habitat use by rock hyraxes (*Procavia capensis*) determined using giving-up densities," *Oikos*, vol. 115, no. 3, pp. 513–525, Dec. 2006.
[57] S. Badenhorst, K. L. van Niekerk, and C. S. Henshilwood, "Rock hyraxes (*Procavia capensis*) from middle stone age levels at Blombos Cave, South Africa," *Afr. Archaeol. Rev.*, vol. 31, no. 1, pp. 25–43, Mar. 2014.
[58] S. Mousavi, A. Mosavi, A. R. Varkonyi-Koczy, and G. Fazekas, "Dynamic resource allocation in cloud computing," *Acta Polytechnica Hungarica.*, vol. 14, no. 4, pp. 83–104, 2017.
[59] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. MGC@ Middleware*, vol. 4, 2010, pp. 1890799–1890803.
[60] M. Bashir, M. A. Latiff, A. Ahmed, A. Yousif, and M. Eltayeeb, "Content-based information retrieval techniques based on grid computing: A review," *IETE Tech. Rev.*, vol. 30, no. 3, pp. 223–232, 2013.

SAURABH SINGHAL received the Ph.D. degree, in 2021. He is currently an Associate Professor with GLA University, Mathura. His research interests include resource management and placement optimization in the field of parallel and distributed systems.

NAKUL GUPTA received the master's degree in the field of construction technology and management, in 2011, and the Ph.D. degree, in 2016. He is currently an Associate Professor with GLA University, Mathura. His research interests include machine learning, geopolymer concrete, and bio concrete.

PARVEEN BERWAL received the master's degree in the field of transportation engineering, in 2011, and the Ph.D. degree, in 2018. He is currently a Professor with the Galgotias College of Engineering and Technology, Greater Noida. His research interests include machine learning, recycled concrete material, and green concrete.

QUADRI NOORULHASAN NAVEED received the Ph.D. degree in information technology from the College/Kulliyyah of Information and Communication Technology (KICT), International Islamic University Malaysia (IIUM), Kuala Lumpur. He was an Engineer with Saudi Aramco and Riyad Bank, Saudi Arabia. He is currently with the College of Computer Science, King Khalid University, Saudi Arabia. His current research interests include e-learning, collaborative computing, cloud-based virtual learning, educational technology, and data science. He has more than 55 publications in SCI/SSCI/Scopus indexed international journals and the IEEE, ACM, Scopus, and Springer-sponsored conferences. He is also a reviewer and an international advisory board of several conferences and SCI/SSCI indexed journals.

AYODELE LASISI is currently an Assistant Professor with the Department of Computer Science, College of Computer Science, King Khalid University, Abha, Saudi Arabia. His research interests include fuzzy logic, artificial intelligence, optimization, and machine learning. He has published good number of articles in national and international journals.

ANTENEH WOGASSO WODAJO is currently an Assistant Professor with the Department of Automotive Engineering, College of Engineering and Technology, Dilla University, Dilla, Ethiopia. His research interests include biodiesel, combustion, optimization, and CRDI. He has published good number of articles in national and international journals.