IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 28, 2021, accepted August 8, 2021, date of publication August 18, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105727

An Energy-Efficient Hybrid Scheduling Algorithm for Task Scheduling in the Cloud Computing Environments

NAVPREET KAUR WALIA[1], NAVDEEP KAUR[2], MAJED ALOWAIDI[3], KAMALJEET SINGH BHATIA[4], (Senior Member, IEEE), SHAILENDRA MISHRA[3], (Senior Member, IEEE), NAVEEN KUMAR SHARMA[5], (Senior Member, IEEE), SUNIL KUMAR SHARMA[3], AND HARSIMRAT KAUR[6,7]

[1]Department of Computer Science and Engineering, Chandigarh University, Ajitgarh, Punjab 140413, India
[2]Department of Computer Science and Engineering, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab 140407, India
[3]Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Al Majma'ah 11952, Saudi Arabia
[4]Department of Electronics and Communication Engineering, Govind Ballabh Pant Institute of Engineering and Technology, Pauri Garhwal, Uttarakhand 246194, India
[5]Department of Electrical Engineering, I. K. Gujral Punjab Technical University, Jalandhar 144603, India
[6]Department of Electronics and Communication Engineering, IKGPTU, Jalandhar, Punjab 144603, India
[7]CTIEMT at Shahpur, Jalandhar, Punjab 144020, India

Corresponding authors: Sunil Kumar Sharma (s.sharma@mu.edu.sa) and Majed Alowaidi (m.alowaidi@mu.edu.sa)

The work of Majed Alowaidi was supported by the Deanship of Scientific Research at Majmaah University under Project R-2021-188.

ABSTRACT The main intent of the cloud computing to provide utilities to the demands of the users that are booming day by day. To meet the requirements, existing scheduling algorithms focus on the improving the performance and neglecting the energy consumed to fulfill those demands. Hence, we propose a new Hybrid Scheduling Algorithm (HS) which is based on the Genetic Algorithm (GA) and Flower Pollination based Algorithm (FPA) for cloud environments. The proposed scheduling algorithm has surpassed in terms of performance across various parameters, i.e. completion time, resource utilization, cost of computation, and energy consumption for both cloud environments than the existing scheduling algorithms (GA and FPA). The simulation results revealed that HS has demonstrated maximum resource utilization with minimum energy consumption in less completion time for the execution of the tasks as compared to the existing scheduling algorithms in both environments. The simulation results have shown that HS has utilization of the resources, 36% better than GA and 16% better than FPA in homogeneous environment whereas in heterogenous environment, HS has performed 12% better than GA and 3.8% better than FPA. The performance of HS has an improvement of 2.6% from FPA and 6.9% from the GA for completion time in homogeneous environment whereas the completion time of the HS is reduced by 17.8% from FPA and 33.7% from GA in heterogeneous environment. For energy consumption, HS has improved 22% than FPA and 11% from GA in the homogeneous environment and HS is 4% better than FPA and 14% from GA in heterogeneous environment.

INDEX TERMS Cloud computing, energy, resource, scheduling, tasks.

I. INTRODUCTION

Cloud computing is the emerging paradigm that renders hardware as well as software as a service, which is commoditised and delivered in a way similar to utilities such as telephony, electricity, gas, and water [1]. It is a simple pay-per-use consumer-provider service model. According to National Institute of Standards and Technology (NIST), 'Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable resources (e.g., networks, servers, storage, applications, and power) that can be rapidly provisioned and released with minimal management effort or service provider interaction'' [2]. For example, Apple iCloud is the service provided by the

The associate editor coordinating the review of this manuscript and approving it for publication was Lo'ai A. Tawalbeh.

enterprise for the ease of clients so that users can access their data stored in the cloud from any device they connect to [3]. The cloud provides a platform for multi-tenant computing which allows the users to share resources. Cloud computing resources are concentrated in a manner to support the varied demands of clients, thus bringing in more opportunities and challenges for Cloud Service Providers [4].

Resource Management is always a significant issue in cloud computing, like any other computing paradigm. Due to the availability of finite shared resources, it is very challenging for cloud providers to fulfil all the requests. Resource Management System deals with the optimum utilisation of shared resources. Manvi and Krishna Shyam [5] stated resource as "any physical or virtual component of limited availability within a computer system. Every device connected to a computer system is a resource." According to Jennings and Stadler [6], the resources can be computing resources, networking resources, storage resources, and power resources. Computing resources are the collection of the physical machines, each comprised of processors, memory, etc. Each physical machine hosts virtual machines that may have different configurations depending on the user requirement. Networking resources consider network topologies, latency, and bandwidth for resource management. Storage resources provide storage services, assuring consistency, and reliability. Power resources focus on energy usage. Various resource management techniques have been proposed in the literature [4], [6], [3] to handle these resources. Resource Management techniques classify as Resource Discovery, Resource Scheduling, Resource Allocation, and Resource Monitoring.

Scheduling algorithm plays a vital role in the resource management system [4]. The scheduling algorithm is responsible for allocating system resources to various tasks submitted by clients/users. While assigning the resources, sometimes the resources are under-utilised or over-utilised, which affects the performance of the scheduling algorithms. According to recent studies by Endo *et al.* [7], the energy consumed by idle resources is much higher as compared to the resources utilised by the data centres in cloud computing which is in a ratio of 60:20 respectively. Also, server energy consumption is directly proportional to the resources used. All over the years, task schedule is a significant research area in different architectures and environments starting from single processor to cloud computing. Cloud computing is a model for enabling ubiquitous network access to a shared pool of configurable computing resources where available resources must be checked and scheduled using an efficient task scheduling algorithm to be assigned to clients.

Traditional resource management techniques are not adequate for cloud computing as they are on virtualisation technology with distributed nature. Cloud computing introduces new challenges for resource management due to heterogeneity in hardware capabilities, on-demand service model, pay per use model and guarantee to meet QoS [3], [6]–[10]. From the cloud providers perspective, allocation of cloud resources must be fairly and efficiently.

So, it is a vital issue to meet cloud consumers QoS requirements and satisfaction like efficient utilisation of resources, cost reduction from cloud provider's perspective, energy reduction [10]–[13].

In this paper, we propose a new scheduling algorithm named as Hybrid Scheduling Algorithm (HS) based on Flower Pollination based Algorithm (FPA) and Genetic Algorithm (GA). FPA and GA are meta-heuristic scheduling algorithms having the significant ability to imitate the best features in nature which help in improving the performance of the scheduling algorithm. The objective is to design an energy-efficient scheduling algorithm for cloud computing that satisfy user's requirements while offering excellent performance.

Meta-heuristic scheduling algorithms provide better and optimal solutions as they are based on the selection of the fittest for finding an optimal solution. Various research [14], [15] proves that the efficiency of task scheduling can be made better manageable with the help of Flower Pollination based Algorithm (FPA). FPA helps in providing optimal solution due to its convergence nature which results in more utilizing of resources with less energy usage. The main drawback of FPA is the time consumed to execute tasks. Hence, we combine some properties i.e. Crossover and Mutation of Genetic Algorithm (GA) with FPA resulting into Hybrid Scheduling Algorithm (HS) which efficiently handles all the tasks and resources assigned in the cloud computing environments. This hybrid combination is not yet implemented in the cloud computing environment as per best of our knowledge and hence this novel approach is considered for providing better services.

Hence, the main motive of this paper is to propose an energy-efficient scheduling algorithm which assigns all the available resources efficiently for the execution of tasks in cloud computing environments.

This paper is divided into 8 Sections. Section II discusses about the related work done in the literature. Section III deliberates about the mechanism of Hybrid Scheduling Algorithm. The simulation environment and performance metrics that are considered is present in Section IV. Furthermore, the results and discussions were analyzed in the next section. The problem of fairness is explained in Section VII and Conclusions were made in the last section with the future scope.

II. RELATED WORK

In this techno-savvy world, the massive demand of the Internet and its services are changing the way we work in our daily routine. Cloud computing is a technology that uses the Internet and unites the world by providing its different services. According to the Berkeley Report [16], "Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service". Cloud computing service providers offer infrastructures, platforms as well as software services to the organizations. With the increasing demand of these services, cloud computing service providers like Amazon AWS, Adobe, Google, Microsoft, Accenture, Cisco and IBM are proactively deploying data centres around the world to deliver Cloud computing services [17].

For the effective realization of potential, cloud computing service providers are bringing in a lot of flexibility and diversification to meet requirements of their clients, which does not involve any intervention or disturbance at client's end in context to infrastructure [17]. The pace of research towards cloud computing is growing fast as observed over the last decade, which has further led to worldwide participation in workshops, conferences and an increase in publications on Cloud Computing [6]. Fakhfakh *et al.* [18] and Zhang *et al.* [9] have presented an overview of cloud computing and focused on the state-of-the-art research challenges in which resource management system is of great importance in the cloud computing environment. According to the survey done by Jennings and Stadler [6], Resource management is very challenging due to the scale of modern data centres; the heterogeneity of resource types and their inter-dependencies; the variability and unpredictability of the load; as well as the range of objectives of the different actors in a cloud ecosystem.

Due to the multifarious nature of resources and uncertainty in the flow of tasks, the level of difficulty increases in context to scheduling in resource management system [4], [19]. The scheduling algorithm maps numerous distinct tasks to the shared resources in the cloud computing environments. During the mapping, some resources remain idle, whereas some exhaust more. It results in underutilization and over-utilisation of the resources. Thus, the management of the resources is of great concern for the scheduling algorithm.

Recent studies [20], [21], have focused on one parameter for the performance evaluation of the task scheduling algorithm. Khanghahi and Ravanmehr [22] stated that many factors, such as workload, usability, location, etc. affect the performance of the task scheduling algorithm. Thus, the performance evaluation of the scheduling algorithm based on a single parameter is not that effective. Hence, this thesis considers four parameters for assessment of the performance of the proposed algorithm.

Nowadays, the increasing demand of services provided by cloud computing has led to various challenges. The heat generated by data centre servers, for attaining efficiency in cloud computing is on the rise. The electricity required for running and for cooling off the data center servers is very expensive because of a lot of energy consumption.

According to Lee and Zomaya [23], the energy consumption issue has undoubtedly improved to a certain level due to the recent advances in hardware technologies. But, because of the different usage patterns of auxiliary and computing resources affecting the amount of energy consumed, has further led sustainable computing into a serious concern [24], [25]. It means that a higher volume of energy consumption is affected by over-utilisation or underutilization of resources as compared to the situation in which resources are effectively utilized. It calls for the development of an efficient scheduling algorithm which allocates each task to the resource by which energy consumption is minimum without performance degradation for the execution of tasks.

Nevertheless, the performance and the scheduling of the tasks to the resources by the scheduling algorithm is very much dependent on cloud computing environments. According to [26], different cloud computing environments are developed to match the varying requirements of clients which are classified as federated and non-federated, static and dynamic, homogeneous, and heterogeneous [27]–[29]. Therefore, this paper takes into consideration two environments, namely homogeneous and heterogeneous.

Kansal and Chana [30] have presented an energy-aware resource utilisation technique based on artificial bee colony (ABC) optimization. An energy-aware model has been developed to enhance resource utilisation and efficiently manage cloud resources. Different workloads such as CPU-intensive workloads and memory-intensive workloads have been considered by the author, as different types of workloads do hinder in virtualized environments. There are two drawbacks in this technique. First, this technique was implemented only in a homogeneous environment. Secondly, they consider only two different workloads.

According to [31], the temperature on physical machines affects the working of cloud computing. As the author discusses, VM creation depends on the temperature of the physical machine. The increase in temperature of PMs results in many environmental hazards. The author proposed a new proactive technique in which if the temperature reaches the threshold value, then cooling of the physical machine is the foremost requirement, and VMs created accordingly.

According to [32], the allocation of VMs to PMs is handled by PSO with the multi-objective fuzzy method. In this, the author concentrates on power consumption, processing resource wastage, and temperature metrics based on which the proposed algorithm outperforms. The main goal of this proposed method is efficiently obtaining a near-optimal solution that minimises the performance parameters.

Mhedheb *et al.* [33] proposed a load-aware and thermal–aware VM scheduling mechanism in which the VM migration is done in such a way that load and temperature are balanced. It also helps in reducing energy consumption. With this technique, physical hosts are relieved from high temperatures as well as over-utilisation. This mechanism implemented in the CloudSim tool.

According to Lee and Zomaya [23], the energy consumption issue has undoubtedly improved to a certain level due to the recent advances in hardware technologies. But, because of the different usage patterns of auxiliary and computing resources affecting the amount of energy consumed, has further led sustainable computing into a serious concern [24], [25]. It means that a higher volume of energy consumption is affected by over-utilization or

underutilization of resources as compared to the situation in which resources are effectively utilized. It calls for the development of an efficient scheduling algorithm which allocates each task to the resource by which energy consumption is minimum without performance degradation for the execution of tasks.

Nevertheless, the scheduling of the tasks to the resources by the scheduling algorithm is very much dependent on cloud computing environments. According to [26], different cloud computing environments are developed to match the varying requirements of clients. Different cloud environments depend upon on preferences of clients such as federated and non-federated, static and dynamic, light-loaded and heavy-loaded, homogeneous, and heterogeneous as displayed in Table 1. The performance of the scheduling algorithm depends on cloud computing environments [27], [28], [29]. Therefore, this paper takes into consideration two environments, namely homogeneous and heterogeneous.

TABLE 1. Existing scheduling algorithms.

Year	Title	Parameters	Environment
2016	Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment	Completion time, execution cost, speedup, resource utilization	Homogeneous
2014	Performance Evaluation of Task scheduling with priority and non-priority in cloud computing	Load Balancing parameter	Homogeneous
2016	Scheduling Jobs on Cloud Computing using Firefly Algorithm	Execution Time	Homogeneous
1999	Solving the Parallel Task Scheduling Problem by Means of a Genetic Approach	Makespan	Homogeneous
2018	A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing	Makespan	Homogeneous
2017	A Flower Pollination Algorithm Based Task Scheduling in Cloud Computing	Makespan, Average cloud utilization	static
2018	A Multi-Objective Task Scheduling Algorithm for Heterogeneous Multi-Cloud Environment	Makespan, Average cloud utilization, Total cost	Heterogeneous

Based on the literature, it becomes necessary to handle the scheduling of the tasks efficiently. Optimization criteria such as minimum completion time, efficient resource utilization, cost-efficiency, maximum throughput, etc. are used to achieve optimal task scheduling [34], [35]. Due to the enormous tasks that are to be scheduled in different cloud environments, it becomes challenging to find the best solution. So, this paper focuses on more than one parameter and considers completion time, energy consumption, resource utilization, and cost of computation as performance metrics for homogeneous and heterogeneous cloud environments.

Two Algorithms are used which are explained as follows:

1. Genetic Algorithm

Genetic algorithm is an evolutionary algorithm which comes under meta-heuristic scheduling algorithms. Genetic Algorithm (GA) has brought an incredible impact to many growing areas, such as Product Process Design, Knowledge discovery, Decision Analysis, Image Processing, Artificial Intelligence, Stock market Analysis, Pattern Recognition, and Resource Scheduling. Due to the multipurpose features of the genetic algorithms, such as easy interfacing with the existing simulations and models, this algorithm has rapidly developed and widely used. A GA comprises chromosomes which signify the complete solutions, i.e., resource allocations. Chromosomes contain a complete set of genes. A gene represented as a task is the underlying data structure of GA and the process of GA. The genetic algorithm consists of the following five steps:

- The first step is the Initialisation step in which the initial population is chosen randomly, generating the entire range of possible solutions (the search space), which are encoded into fixed binary strings. In this step, the resource population and task population is generated randomly.
- The second step is Fitness Evaluation, in which the solution obtained from step-1 is used to evaluate the fitness values. The fitness function is the method of measuring the quality of the represented solution. Here, based on the potential of the resources and requirements of the tasks, fitness is calculated.
- The third step is the selection of the population. In this step, based on the fitness values, the best solutions are selected. With the genetic material, better ones survive to the next generation passed on, which means for its execution best resource is selected for the best-matched task.
- The fourth step is applying genetic operators, namely, Crossover and Mutation. In this step, with the use of the combination of genetic operators, the possible second-generation population of solutions is generated. This method continuously makes the population until a new population of

appropriate size is generated. For a crossover, possible generated solutions are matched to for new generation, and mutation, allocations are swapped.

- The last step is the Termination step. In this step, when all jobs are executed entirely, then it stops. This execution is continuous until when a fixed number of generations with a solution that satisfies minimum criteria is achieved.

2. Flower Pollination Algorithm

Flower Pollination based algorithm is a bio-inspired scheduling algorithm based on the concept of flower pollination. Flower pollination is a procedure that acts within nearly 80 per cent of the earth's plant species. Flower pollination is a process where pollinators such as insects, birds, bats, other animals take care of the transfer of pollen grain. For a successful pollination process, some flowers can only attract a specific species of insect or bird, and there are only two primary forms of the biotic and a-biotic pollination process. In the cycle of biotic pollination, pollen grains are carried by insects and animals from the pollinator. In the process of a-biotic pollination, they do not want a pollinator. This flower pollination phenomenon can be used to solve several distributed and complex computational issues of the cloud computing world. It also helps us to address resource management problems in the cloud computing system. The steps for algorithms as follow:

- The first step is Initialisation in which a random population of n flower/pollen gametes is initialised with random solutions. When scheduling is done, flowers are taken as resources, and pollen are taken as tasks in the cloud systems.
- The second step is the Selection of the Best Solution (g^*) in which the best solution is selected, i.e., the best resource from the generated population and also afterwards define the switching probability between 0 and 1.
- The third step is to Evaluate New Solutions. While scheduling, for each iteration, if rand is less than switch probability, then global pollination is executed; otherwise, local pollination is implemented, and then the new solution is evaluated accordingly.
- The fourth step is to Update population. It means that if the new best solution is not optimal than the g^*, then the population is updated; otherwise, the stop criterion is checked.
- The fifth step is the Termination in which the execution is done when a fixed number of generations satisfy minimum criteria with an optimal solution. It means when all jobs are executed, then it halts.

III. HYBRID SCHEDULING ALGORITHM

Hybrid-Scheduling Algorithm (HS) comprises of the Flower Pollination-based Algorithm (FPA) [15] and Genetic Algorithm (GA) [36] to assign resources to their tasks. In this scheduling algorithm, we take tasks and resources as inputs in the form of flowers and pollens. A random population generated with the combination of all resources and tasks. The processing of the FPA is applied to the resources to find out the VMs which can solve the tasks assigned by the clients. After processing, the output of the pollination is taken as input for the Genetic algorithm. One point crossover and swapping method of mutation is used in this approach. It provides the best-matched solution as per the requirement. HS helps the users to ensure efficient machines to execute their tasks and better performance for handling multiple tasks at the same time. Hence, HS is an energy-efficient algorithm which distributes the resources among tasks with less energy. Also, the complexity of the proposed algorithm is O(n). In this cloud computing model, 'n' number of independent tasks represented as T_1, T_2, $T_3 \ldots\ldots T_n$ assigned to 'k' virtual machines described as VM_1, VM_2, $VM_3, \ldots\ldots VM_k$ for the execution. These virtual machines generated from 'm' physical machines called cloud resources represented as R_1, R_2, $R_3, \ldots\ldots$ Rm for both homogeneous and heterogeneous cloud computing environments. This mapping is based on specific parameters. All these entities have their characteristics. Like, a task T_i identified by its unique ID expressed as T_{ID} and name denoted as T_{Name}. Every task contains information like a set of instructions means the length of the task T_{Length}, Date and time T_{date}, the identity of the user T_{user} who submitted the task and requirement of the task $T_{Require}$ that defines the category of the task is represented as:

$$T_i = f(T_{ID}, T_{Name}, T_{Length}, T_{date}, T_{user}, T_{Require})$$

Similarly, A Resource Rj identified by its ID and Name represented as R_{ID} and R_{Name} respectively. Each resource contains information like memory R_{RAM}, Current Temperature R_{Temp}, Operating System R_{OS}, Processor $R_{Processor}$, and Storage Space R_{Space} expressed as:

$$R_j = f(R_{ID}, R_{Name}, R_{RAM}, R_{Temp}, R_{OS}, R_{Processor}, R_{Space})$$

Virtual Machines are vital part of this model and identified using its unique ID VM_{ID} and Name VM_{Name}. It contains information like, the VM belongs to which resource by using R_{ID} and the capability of the machine to handle the task using its MIPS rating VM_{MIPS} and is represented as:

$$VM_k = f(VM_{ID}, VM_{Name}, R_{ID}, VM_{MIPS})$$

The working process of the Hybrid Scheduling Algorithm (HS) shown in Figure 1 and described as given below is explained in following phases:

- Input

In this phase, firstly we initialize population size according to the task and resource requirements and maximum number of generations (maxgen) which is 20. Then, switch probability P is defined whose valu lies between [0,1]. Then, the population of tasks taken as flowers and resources taken as pollens is initialized with the random solution(rand).

To maintain the competence of the Hybrid Scheduling algorithm, VM is generated based on the criterion which is checked by the VM generator, as shown in Figure 2. Here, 'n' number of physical resources generates 'm' virtual machines depending on the criterion passed to VM Generator

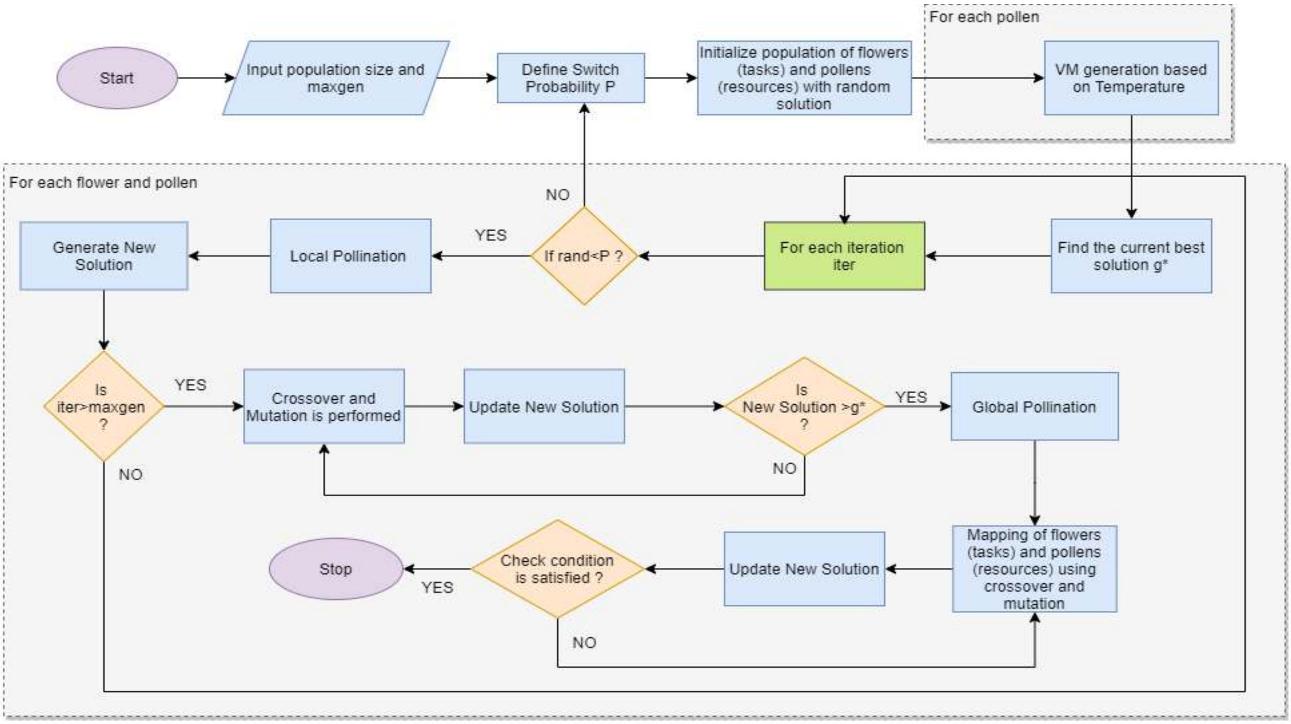

Start
Input population size and maxgen
Define Switch Probability P
Initialize population of flowers (tasks) and pollens (resources) with random solution
For each pollen
VM generation based on Temperature
For each flower and pollen
Find the current best solution g*
For each iteration iter
If rand<P ?
NO
YES
Local Pollination
Generate New Solution
Is iter>maxgen ?
YES
NO
Crossover and Mutation is performed
Update New Solution
Is New Solution >g* ?
YES
NO
Global Pollination
Mapping of flowers (tasks) and pollens (resources) using crossover and mutation
Update New Solution
Check condition is satisfied ?
YES
NO
Stop

FIGURE 1. Design of hybrid scheduling algorithm.

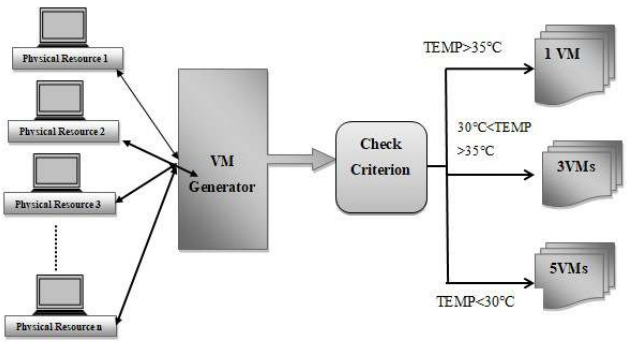


FIGURE 2. VM generation based on temperature for cloud computing.

where $m>n$. This mechanism generates a Virtual Machine (VM) from a Physical Machine (PM) based on its temperature. Here, the criteria comprise of three different temperature conditions that are set for VM Generation. Three different categories are as follows: (a) Very High (VH), (b) High (H), and (c) Low (L). If the temperature of the PM is higher than VH (>35°C), then only one VM is generated, which means that one device works like two devices only. If the temperature is between H and L (<35°C and >30°C) and under L (<30°C), then three and five VMs are generated from single resources, respectively, as shown in Figure 2. Hence, VMs are generated based on the temperature for each pollen and the current best solution (g*) is found. The temperature of the physical resources are considered before the execution of the tasks and during execution, the temperature varies.

Algorithm: Hybrid Scheduling Algorithm (HS)

- Local Pollination

In this phase, the value of rand is checked. If the value of rand is less than P, then Local Pollination is performed for both flowers and pollens. In this pollination, flowers (tasks) and pollens(resources) are separately sorted using uniform distribution and new solution is generated. If all the tasks and resources are sorted then swapping method of mutation and one point crossover operators of the Genetic Algorithm are performed and new solution is updated. This new solution is checked whether it is greater than g* or not. If yes, then it is passed for global pollination otherwise crossover and mutation operators are performed again.

- Global Pollination

In this phase, global pollination is performed on the updated new solution using Levy flights distribution. Mapping of sorted flowers (tasks) and sorted pollens (resources) are performed in this phase. One point crossover and swapping method of mutation operators are again used for generating best solutions until all flowers (tasks) are mapped with the pollens (resources).

```
Data: a set of flowers(tasks), a set of pollens (resources)
Result: Finding the best mapping of flowers with pollens
$Task \leftarrow set\ of\ tasks$
$Resource \leftarrow set\ of\ resources$
Input population size, maxgen;
Input Resource $R_{id}$, $R_{name}$, $R_{RAM}$, $R_{Temp}$;
Input Task $T_{id}$, $T_{name}$, $T_{desc}$, $T_{inst}$, $T_{req}$;
Define switch probability P $\epsilon$ [0,1];
```

```
9  Initialize population of flowers(tasks) and
   pollens(resources) with random solution;
10 $rand_{pop}$ ←population of flowers/pollens gametes with
   random solutions
11 for (each pollen ϵ Resource)
12         if ($R_{Temp} > 25°$ and $R_{Temp} <= 30°C$)
13             5VM's← VM;
14         end if
15         if ($R_{Temp} > 30°$ and $R_{Temp} <= 35°C$)
16             3VM's ←VM;
17         end if
18         if ($R_{Temp} > 35°C$)
19             1VM ←VM;
20         end if
21 end for
22 for (each pollen ϵ Resource and each flower ϵ Task)
23         Find the current best solution g*;
24         for (each iteration iter)
25             if ($rand_{pop}! < P$)
26                 Go to Step 8 and define switch
                   probability P again;
27             end if
28             Draw from Uniform Distribution in [0,1]
               and perform Local Pollination;
29             Generate New Solution;
30             if (iter!>max)
31                 iter=iter+1;
32             end if
33         end for
34         Perform crossover and mutation;
35         Update New Solution;
36         If (New Solution!>g*)
37             Go to Step 34 and perform crossover and
               mutation again;
38         end if
39         Draw from Levy flights distribution and
               perform Global Pollination;
40         Mapping of flower and pollens using
           crossover and mutation;
41         Update New Solution;
42         if (condition not satisfied)
43             Go to Step 40 and perform mapping
               again;
44         end if
45 end for
```

IV. SIMULATION ENVIRONMENT AND PERFORMANCE METRICS

We consider ASP.Net as the simulation tool for our Experiments. Two different cloud computing environments, namely, homogeneous and heterogeneous, are created with the help of the simulation tool, as displayed in Table 3. Here, in the cloud computing environment, one data centre is considered which contains 100 physical resources with their own configurations.

The proposed algorithm, HS, uses one threshold temperature (High) for all the resources in the homogeneous environment. In contrast, we use three threshold temperatures (Low, High and Very High) for the heterogeneous environment. The generation of VMs depends on these threshold temperatures as already discussed in the above section. Initially, random temperature is allocated to machines which changes while executing the tasks. We took different workloads as displayed in Table 2 to analyze the performance of HS and compares with GA and FPA on the basis of performance metrics in the next section. These workloads or assigned tasks has three different types of resource requirement i.e. CPU-intensive, Memory-intensive or Both.

TABLE 2. Number of workloads.

Workloads	Number of Assigned Tasks
Workload_1	100
Workload_2	200
Workload_3	300
Workload_4	400
Workload_5	500
Workload_6	600

In our experiments, we will be using the following metrics to scrutinize the performance of the meta-heuristic scheduling algorithms:

- Resource Utilization (RU)

Resource utilization is an essential factor in analyzing the performance of physical machines in terms of handling resources. So, it defines the number of resources utilized for the execution of a given amount of tasks. The calculation of the resource utilization factor is as follows:

$$Resource\ Utilization\ (in\%) = \sum_{i=1}^{i=n} Number\ of\ tasks\ (K_i)\ executedper\ resource(R_i)$$

- Completion Time (CT)

Completion Time defines the amount of time required to complete the task. The task is a set of instructions that can be handled by a physical machine where the physical machine has its rate of handling the instructions called (Million Instruction per Second) MIPS rating. Completion Time is the sum of Execution time and Waiting Time and calculates as follows:

$$Completion\ Time\ (in\ sec) = \sum_{i=1}^{i=n} (T_i + WT_i)$$

where CT = Completion Time of the allocated tasks,

WT_i = Waiting time of the task for the availability of the resource and

T_i = Execution Time of the allocated task

TABLE 3. Simulation environment specification.

Parameters	Value in Homogeneous Environment	Value in Heterogeneous Environment
Data centre	1	1
Number of Resources	1-100	1-100
Number of VMs	300	300-500
RAM	4GB	1,2 and 4GB
Number of Assigned Tasks	100,200,300,400,500,60 0	100,200,300,400,500,60 0
MIPS(PMs)	6000	2500-10000
Temperatur e	Fixed	Variable (Low, High, Very High)
Space	1TB	500GB and 1TB
Processor	Quad-core	Quad and Octa-core
Task Length	1000-15000MI	1000-15000MI
OS	Linux OS	Windows and Linux OS

Now, Execution Time (ET) defines the time taken by the physical machines for the execution of the task. So, for calculating the time for the execution of the tasks by physical machine is by using the following formula:

$$Execution\ Time(T_i) = \frac{Number\ of\ instructions\ (task(i))}{MIPS\ rating\ (Machine\ \ (i))}$$

The above formula uses to calculate the execution time for only one task. Waiting Time defines the total amount of time taken by the task before execution. Waiting Time depends on the availability of the resource.

- Energy Consumption (EC)

In the cloud computing environment, the physical machines consume energy for providing the services to the users, and the energy consumption depended on the temperature of the physical machines. Here, the total temperature defined as the sum of the temperature of the total resources utilized in the task execution. In contrast, the average temperature is the average temperature of all the resources used for simulation. Hence, energy consumption is the amount of energy consumed for the execution of the tasks, and the calculative formula is as below:

$$\begin{aligned} &Energy\ Consumption\ (in\ Kj)\\ &= \frac{Total\ Temperature}{Average\ Temperature} * Total\ Number\ of\ resources\ used \end{aligned}$$

- Cost of Computation (CC)

Cost of Computation defined as the cost for simulating the tasks on the cloud computing environment with the help of resources. Therefore, cost of computation determines the usage of physical and virtual machine w.r.t tasks, and it is calculated based on the utilization of the total number of VMs and resources for the execution of the given number of tasks using the following formula:

$$\begin{aligned} Cost\ of\ Computation = \frac{Number\ of\ VMs\ used}{Total\ Number\ of\ resources\ used}\\ * Total\ number\ of\ tasks \end{aligned}$$

V. RESULTS AND DISCUSSION

We evaluate the performance of the proposed Hybrid Scheduling Algorithm (HS) and compare with the performance of Genetic Algorithm (GA) [36] and Flower Pollination based Algorithm (FPA) [15] through simulation approach based on performance metrics. We first discuss the Experiments conducted for Homogeneous Cloud Computing Environment, and then we consider the Heterogeneous Cloud Computing Environment for each performance metric.

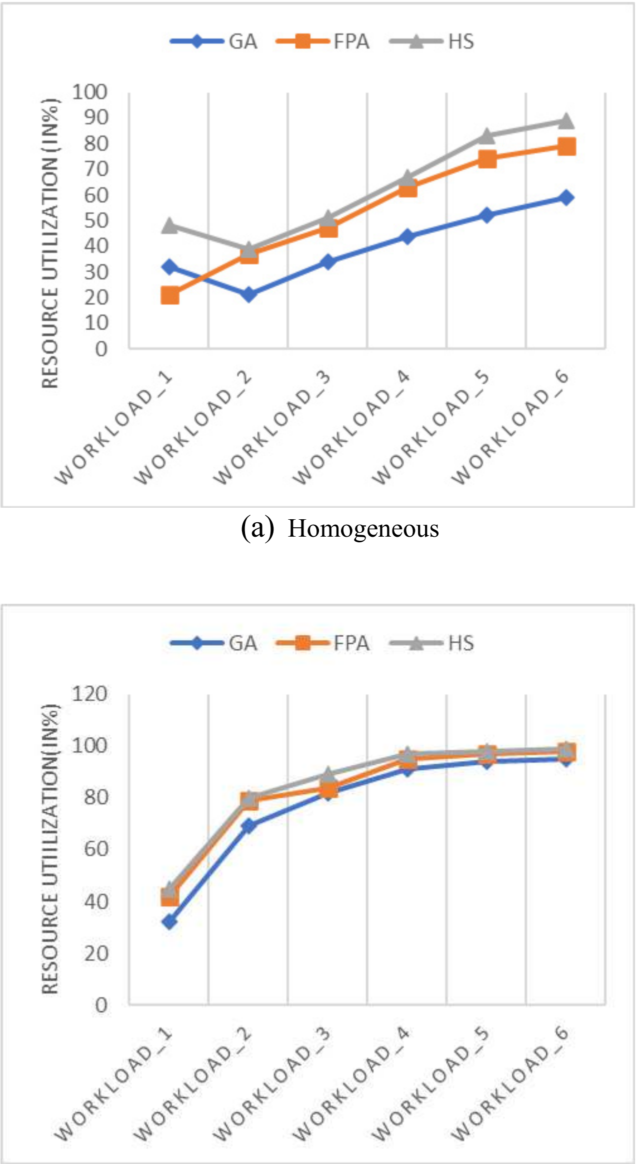
Resource Utilisation: When the scheduling algorithm assigns the tasks to the resources, then at times, some resources remain under-utilized and some over-utilized. There forth, the main focus of the scheduling algorithm is to complete the execution of the tasks by utilizing maximum resources in less time. HS has used more resources than FPA and GA in both cloud computing environments for different workloads.

The graph of Figure 3(a) revealed that in a homogeneous environment, for Workload_1, FPA has used less resource than GA and HS. It means that some resources are idle or some are exhaust more. But when from Workload_2 to Workload_6 is seen, GA has the same condition of using less resource. There is also changeability in the utilisation of resources for Workload_2 in GA. It possibly affected the performance of HS and shown variation in the results of Figure 3(a). Results have also shown that HS has utilized resources 36 % better than GA and 16 % better than FPA.

In the heterogeneous environment, resources have different configurations, and disadvantage is the availability of the resource is not guaranteed. But the advantage of the heterogeneous environment is heterogeneous resources used for the completion of the task, which means maximum resources are used in this environment as compared to the homogeneous environment. 98 % of the resources are maximum utilized as displayed in the graph of Figure 3(b). HS achieves this percentage for Workload_5, and when the case is for Workload_6, FPA has also made this percentage. It means that HS has effectively used the resources for the completion of the tasks. The results revealed that HS had performed 12% better than GA and 3.8% better than FPA in utilizing the resources.

Completion Time: For any scheduling algorithm, the completion time depends on the utilisation of the resources.

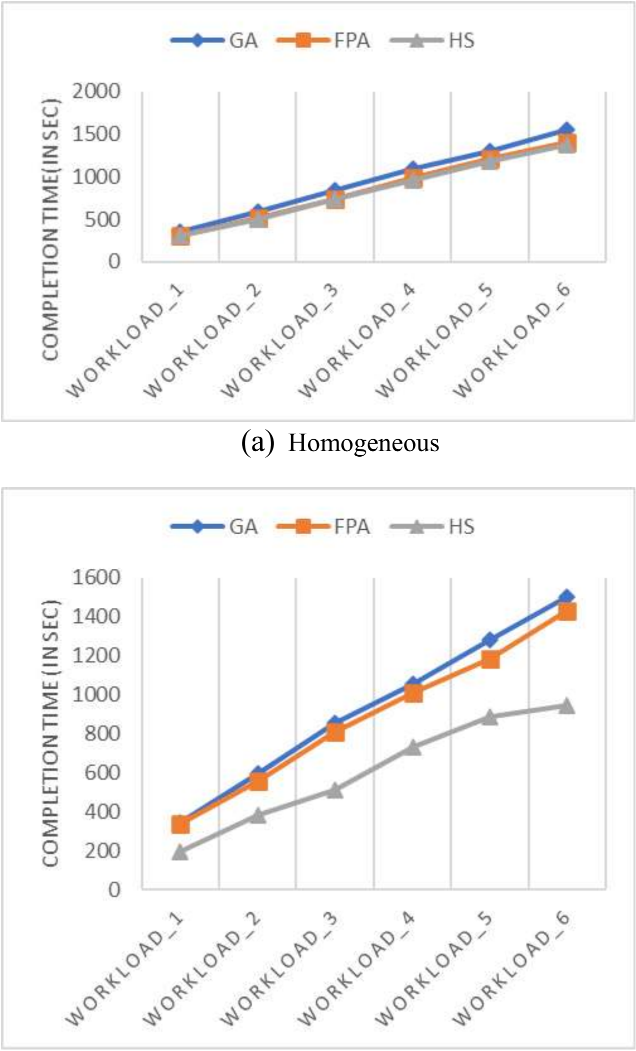
The graph in Figure 4(a) depicts the performance of HS, GA and FPA scheduling algorithms in a homogeneous environment in terms of completion time. Results have shown that there is less variation in the values of the completion time between GA, FPA and HS. It is because of the homogeneous resources used by the scheduling algorithms where all resources have the same configuration. The performance



(a) Homogeneous

(b) Heterogeneous

FIGURE 3. Resource utilization (in %) in cloud computing environments.



(a) Homogeneous

(b) Heterogeneous

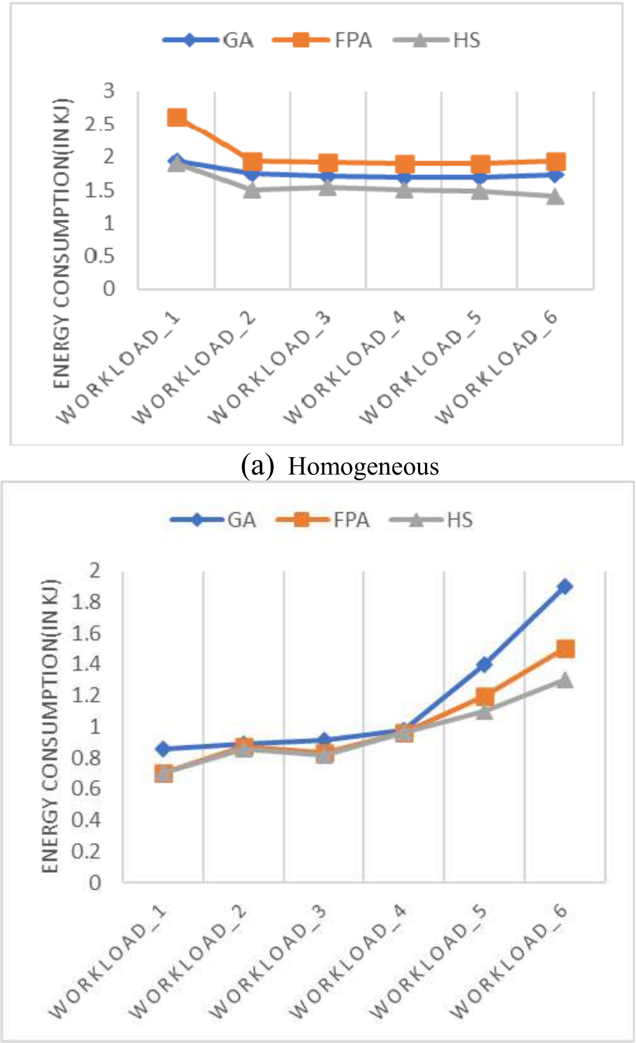
FIGURE 4. Completion time (in sec) in cloud computing environments.

of HS has an improvement of 2.6% from FPA and 6.9% from the GA, as seen in Figure 4(a).

The graph in Figure 4(b) shows that remarkable effect in the performance of HS when compared with GA and FPA in a heterogeneous environment. The generation of the VMs based on three threshold temperatures of the PM has propelled HS for the completion of the tasks in lesser time. As results have shown, the completion time of the HS is reduced by 17.8 % from FPA and 33.7 % from GA.

Energy Consumption: Nowadays, the scheduling algorithm focuses not only to provide better services to cloud users by performing the execution of tasks efficiently but also in maintaining energy usage. As seen in a homogeneous environment, FPA has consumed maximum energy, whereas, in a heterogeneous environment, GA has maximum energy consumption. It means that when we use similar resources, FPA has to consume more energy to find an optimal solution. Still, when we use different resources, GA has wasted more energy for the execution of tasks. The results also revealed that HS had consumed less energy than GA and FPA for both cloud computing environments because, in Hybrid Scheduling Algorithm (HS), FPA is used to assign resources to tasks with the help of GA.

The graph in Figure 5(a) illustrates the energy consumption of GA, FPA and HS for the homogeneous environment. In the case of Workload_1, when the number of tasks is less than the energy consumed by GA is very high, whereas there is very less difference between the values of FPA and HS. As the number of tasks increases, the difference between FPA and HS also increases. It is because HS is a combination of FPA and GA, and the performance of FPA and GA has some effect



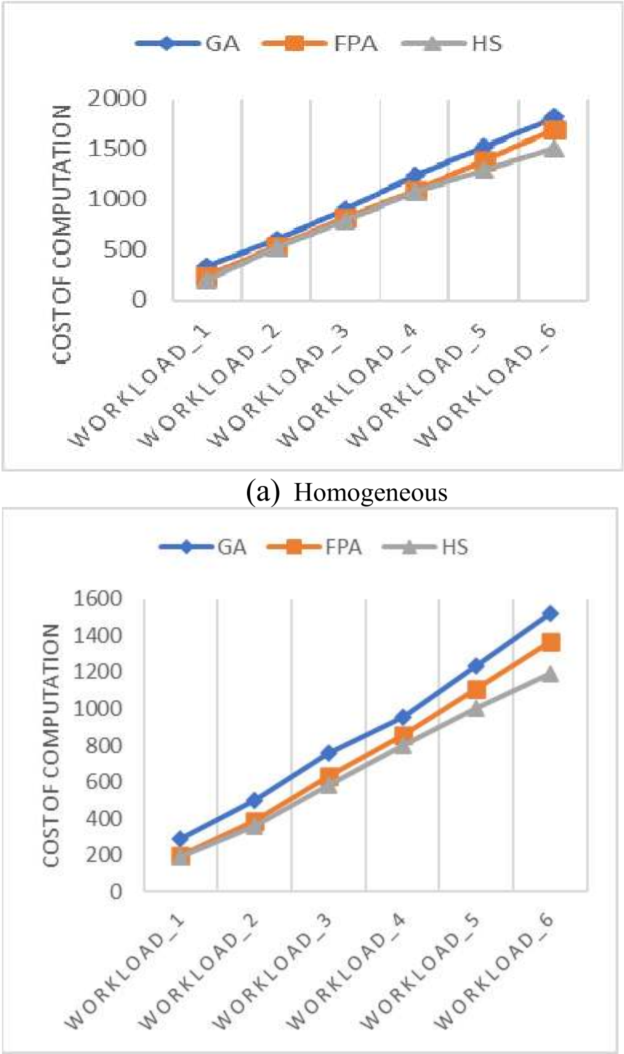
(a) Homogeneous

(b) Heterogeneous

FIGURE 5. Energy consumption (in Kj) in cloud computing environments.

on HS. The results shown in Figure 5(a) depict that Hybrid Scheduling Algorithm (HS) has improved 22% than FPA and 11% from GA in the homogeneous environment. When the energy consumption for the heterogeneous environment is analyzed as displayed in the graph of Figure 5(b), from Workload_1 to Workload_4, the difference in the use of energy between FPA and HS is very less. GA also has less difference when the case is Workload_2 and Workload_4. After this, there is a tremendous increase in the energy consumption for Workload_5 and Workload_6 by GA and FPA as compared to HS. These fluctuations can be because of the heterogeneous resources that we use for the execution of the tasks. As shown in the graph of Figure 5(b), HS is 4 % better than FPA and 14% from GA in terms of energy consumption.

Cost of Computation: Cost of Computation depends on the number of resources used for the completion of the tasks. If the utilisation of the resources is more than the cost of computation will be less.



(a) Homogeneous

(b) Heterogeneous

FIGURE 6. Cost of computation in cloud computing environments.

The graph of Figure 6(a), reveals that HS has an improved performance than FPA and GA in terms of cost of computation in a homogeneous environment. When Workload_1 considered, HS has an improvement of 13.9% than FPA, which is maximum, and when we consider Workload_4, HS has a gain of only 1.4% than FPA, which is minimum. The results of the graph of Figure 6(a) also revealed that GA has the cost of computation 12% higher than FPA and 18% than HS, which means it has overall less resource utilisation. The graph of Figure 6(b) depicts the performance of the HS. HS outperforms GA and FPA in terms of the cost of computation in the heterogeneous environment. As seen in the graph of Figure 6(b), GA and FPA have fluctuations in the different workloads, and HS has consistently improved in the performance from Workload_1 to Workload_6. The overall improvement of the HS as compared to FPA and GA is 7.9% and 19.9% respectively.

VI. FAIRNESS PROBLEM

Performance and fairness of the scheduling algorithms are mainly studied independently [37]. Most of the task scheduling algorithms for achieving the optimal performance adheres to the maximum resource utilization. But, according to [37], the scheduler must guarantee two aspects: one is good performance in terms of resource utilization and completion time, and second one is fairness as an important factor to keep user satisfaction. Unfairness in task scheduling occurs when the scheduling algorithm is unable to handle computing requirements during the execution of the tasks. This leads to starvation in the task execution. A very good example of Carpool problem is taken by [38], [39] for explaining fairness in scheduling. Due to unfairness, some tasks may be starved at the expense of others and starvation is not adequately expressed by the performance metrics. A good performance does not guarantee fairness in terms of starvation. Hence, an effective scheduler must have an efficient scheduling algorithm which needs to be fair with good performance [37]. Task scheduling algorithms has been studied extensively but the issue of fairness in task scheduling has usually not been considered much quantitatively [39], [40].

Fairness can be achieved when the scheduling algorithm equally distributes all the resources among various tasks in cloud computing. However, according to [41], [42], the fairness cannot be achieved by simply distributing an equal amount of computing resources to every task. Some task may prefer the resources which have less waiting time, while other task may prefer the resources which provides less completion time. Also, it is possible that a task is allocated preferred resources, but the resources are utilized by multiple tasks that degrade the performance of the scheduling algorithm significantly as well as leading to starvation of resources. With the simple fair allocation of the resources, the scheduling algorithm ends up having poor fairness in these cases [43]. Hence, some common fairness metrics such as Measures of dispersion, Fair start-time analysis and the Resource Allocation Queuing Measure (RAQFM) are existing in the literature. Mostly, deviation methods are used to ensure the fairness among resources [43], [44]. We have used Jain's fairness index [45] as fairness metric using the formula below:

$$Standard\ Deviation(SD) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{1}$$

$$Mean(\bar{x}) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2}$$

$$Coefficient\ of\ Variation\ (\hat{c_v}) = \frac{Standard\ Deviation(SD)}{Mean} \tag{3}$$

$$Jain's\ Fairness\ Index(J) = \frac{1}{1+\hat{c}_v^2} \tag{4}$$

where N = total number of resources which is taken as 100

i = number of resources varies from 1 to 100

x = number of tasks per resource

$\bar{x}$ = mean of total number of assigned tasks per resource

According to [45]–[47], the Jain's Fairness Index (J) ranges from 1/N (worst case) to 1 (best case) and it is maximum when all the resources are fairly distributed among the tasks. Standard deviation for tasks per resource is calculated as shown in Equation (1). After calculating SD, Mean is calculated with the formula in Equation (2). Calculation of variation is obtained using Equation (3). There forth, fairness is seen using the Equation (4). For Jain's Fairness Index(J), the middle value (0.5) is considered as threshold value which depicts if the value of J is more than threshold value than the distribution of the resources is towards fairness otherwise if the value is less than distribution of the resources is towards unfairness. We investigate that the Jain's Fairness Index of GA, FPA and HS for different workloads is less than threshold value for all workloads as shown in Figure 7. For example, for Workload_1 is 0.35 whereas for Workload_3, the value is 0.18, and hence, HS depicts unfairness.

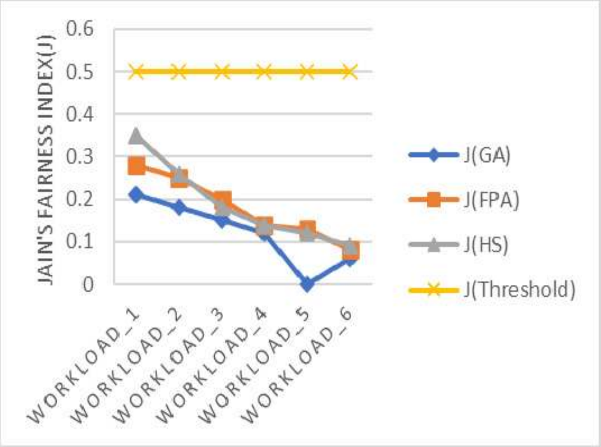


FIGURE 7. Jain's fairness index (J).

VII. CONCLUSION AND FUTURE SCOPE

Efficient task scheduling is one of the main critical challenges in cloud computing. This paper proposed an energy-efficient scheduling algorithm (HS) for task scheduling in different cloud computing environments, which helps to provide available resources to the tasks for execution efficiently. The HS has three phases, out of which two phases are very significant comprising of local pollination and global pollination. The former phase helps in sorting the tasks and resources separately while the latter helps in mapping the sorted tasks to the sorted resources. Both these phases help in scheduling the tasks to available resources competently. The combination of FPA and GA in HS helps in the execution of the tasks by utilizing maximum resources with less energy consumption in less completion time. The simulation results (using ASP.NET) achieved by comparing the HS with other scheduling algorithms like GA and FPA, revealed that the proposed algorithm is capable of producing better and efficiently scheduling tasks and managing resource than the existing scheduling algorithms in different cloud computing environments. However, due to the heterogeneity of workloads poses a significant challenge to HS for fair allocation of the resources for task scheduling in the cloud computing environments. HS has Jain's Fairness Index (J) value less than threshold value, thus exhibits unfairness. The unfairness of the scheduling algorithm occurs when the task is unable to attain the required resources, and the task has to wait for the execution that will increase the completion time. Hence, we extend the proposed HS to include fairness in the distribution of the available resources to the tasks based on the workload type in different cloud computing environments in the future. Also, our performance studies restricted to a maximum of 600 assigned tasks and 100 resources for homogeneous and heterogeneous cloud computing environments. This workload can be extended in the future and can be performed on the real cloud environments in future.

REFERENCES

[1] M. Guazzone, "Power and performance management in cloud computing systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Turin, Turin, Italy, 2012. [Online]. Available: https://dott-informatica.campusnet.unito.it/html/theses/guazzone.pdf

[2] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services," 2009, *arXiv:0903.2525*. [Online]. Available: http://arxiv.org/abs/0903.2525

[3] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, vol. 47, pp. 1–33, Jul. 2015, doi: 10.1145/2788397.

[4] P. Kowsik and K. Rajakumari, "A comparative study on various scheduling algorithms in cloud environment," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 112–121, 2014.

[5] 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, doi: 10.1016/j.jnca.2013.10.004.

[6] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," *J. Netw. Syst. Manage.*, vol. 23, no. 3, pp. 567–619, 2015, doi: 10.1007/s10922-014-9307-7.

[7] P. T. Endo, A. V. de Almeida Palhares, N. N. Pereira, G. E. Goncalves, D. Sadok, J. Kelner, B. Melander, and J.-E. Mangs, "Resource allocation for distributed cloud: Concepts and research challenges," *IEEE Netw.*, vol. 25, no. 4, pp. 42–46, Jul./Aug. 2011, doi: 10.1109/MNET.2011.5958007.

[8] L. Alboaie, S. Alboaie, and A. Panu, "Swarm communication—A messaging pattern proposal for dynamic scalability in cloud," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun. IEEE Int. Conf. Embedded Ubiquitous Comput.*, Nov. 2013, pp. 1930–1937, doi: 10.1109/HPCC.and.EUC.2013.277.

[9] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010, doi: 10.1007/s13174-010-0007-6.

[10] S. Mustafa, B. Nazir, A. Hayat, A. R. Khan, and S. A. Madani, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Comput. Electr. Eng.*, vol. 47, pp. 186–203, Oct. 2015, doi: 10.1016/j.compeleceng.2015.07.021.

[11] D. Sangita, C. Ankita, and P. Reshamlal, "A review on issues and challenges of cloud computing," *Int. J. Innov. Adv. Comput. Sci*, vol. 4, no. 1, pp. 81–88, 2015.

[12] M. Nazir, "Cloud computing: Overview & current research challenges," *IOSR J. Comput. Eng.*, vol. 8, no. 1, pp. 14–22, 2012, doi: 10.9790/0661/0811422.

[13] K. D. Schewe, K. Bósa, H. Lampesberger, J. Ma, M. Rady, and B. Vleju, "Challenges in cloud computing," *Scalable Comput.*, vol. 12, no. 4, pp. 385–390, 2011.

[14] M. Abdel-Basset and L. A. Shawky, "Flower pollination algorithm: A comprehensive review," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2533–2557, 2019, doi: 10.1007/s10462-018-9624-4.

[15] I. Gupta, A. Kaswan, and P. K. Jana, "A flower pollination algorithm based task scheduling in cloud computing," in *Computational Intelligence, Communications, and Business Analytics*, vol. 776, J. K. Mandal, P. Dutta, and S. Mukhopadhyay, Eds. Singapore: Springer, 2017, pp. 97–107.

[16] A. Fox, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Dept. Electr. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS, 2009, vol. 28, no. 13.

[17] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012, doi: 10.1016/j.future.2011.04.017.

[18] F. Fakhfakh, H. H. Kacem, and A. H. Kacem, "Workflow scheduling in cloud computing: A survey," in *Proc. IEEE 18th Int. Enterprise Distrib. Object Comput. Conf. Workshops Demonstrations*, Sep. 2014, pp. 372–378, doi: 10.1109/EDOCW.2014.61.

[19] P. Kumar and S. Anand, "An approach to optimize workflow scheduling for cloud computing environment," *J. Theor. Appl. Inf. Technol.*, vol. 57, no. 3, pp. 1–7, 2013.

[20] M. A. Mondal, S. Choudhury, and M. S. Islam, "Performance analysis of VM scheduling algorithm of CloudSim in cloud computing," *Int. J. Electron. Commun. Technol.*, vol. 6, no. 1, pp. 34–36, 2015.

[21] S. Bilgaiyan, S. Sagnika, and M. Das, "An analysis of task scheduling in cloud computing using evolutionary and swarm-based algorithms," *Int. J. Comput. Appl.*, vol. 89, no. 2, pp. 11–18, Mar. 2014, doi: 10.5120/15473-4158.

[22] N. Khanghahi and R. Ravanmehr, "Cloud computing performance evaluation: Issues and challenges," *Int. J. Cloud Comput., Services Archit.*, vol. 3, no. 5, pp. 29–41, Oct. 2013, doi: 10.5121/ijccsa.2013.3503.

[23] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *J. Supercomput.*, vol. 60, no. 2, pp. 268–280, 2012, doi: 10.1007/s11227-010-0421-3.

[24] C.-H. Hsu, K. D. Slagter, S.-C. Chen, and Y.-C. Chung, "Optimizing energy consumption with task consolidation in clouds," *Inf. Sci.*, vol. 258, pp. 452–462, Feb. 2014, doi: 10.1016/j.ins.2012.10.041.

[25] A. R. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," *Future Gener. Comput. Syst.*, vol. 91, pp. 407–415, Feb. 2019, doi: 10.1016/j.future.2018.09.014.

[26] P. K. Suri and S. Mittal, "A comparative study of various computing processing environments: A review," *Int. J. Comp. Sci. Inf. Technol.*, vol. 3, no. 5, pp. 5215–5218, 2012.

[27] H. S. Narman, M. S. Hossain, and M. Atiquzzaman, "H-DDSS: Heterogeneous dynamic dedicated servers scheduling in cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 3475–3480. [Online]. Available: http://ieeexplore.ieee.org/document/6883859/, doi: 10.1109/ICC.2014.6883859.

[28] G. Da Cunha Rodrigues, R. N. Calheiros, V. T. Guimaraes, G. L. D. Santos, M. B. de Carvalho, L. Z. Granville, L. M. R. Tarouco, and R. Buyya, "Monitoring of cloud computing environments: Concepts, solutions, trends, and future directions," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, Apr. 2016, pp. 378–383, doi: 10.1145/2851613.2851619.

[29] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in *Proc. Int. Conf. Green Comput.*, Aug. 2010, pp. 357–364, doi: 10.1109/GREENCOMP.2010.5598294.

[30] N. J. Kansal and I. Chana, "Artificial bee colony based energy-aware resource utilization technique for cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 5, pp. 1207–1225, Apr. 2015, doi: 10.1002/cpe.3295.

[31] Z. Lee, Y. Wang, and W. Zhou, "A dynamic priority scheduling algorithm on service request scheduling in cloud computing," in *Proc. Int. Conf. Electron. Mech. Eng. Inf. Technol.*, Aug. 2011, pp. 4665–4669, doi: 10.1109/EMEIT.2011.6024076.

[32] R. A. I. Alsini and M. A. Tawfeek, ''Power consumption and temperature minimization in cloud computing,'' *IOSR J. Eng.*, vol. 7, no. 3, pp. 49–62, 2017.

[33] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, and A. Streit, ''Load and thermal-aware VM scheduling on the cloud,'' in *Algorithms and Architectures for Parallel Processing*, vol. 8285, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Kołodziej, B. Di Martino, D. Talia, and K. Xiong, Eds. Cham, Switzerland: Springer, 2013, pp. 101–114.

[34] Q. Duan, ''Cloud service performance evaluation: Status, challenges, and opportunities—A survey from the system modeling perspective,'' *Digit. Commun. Netw.*, vol. 3, no. 2, pp. 101–111, May 2017, doi: 10.1016/j.dcan.2016.12.002.

[35] B. G. Batista, J. C. Estrella, C. H. G. Ferreira, D. M. L. Filho, L. H. V. Nakamura, S. Reiff-Marganiec, M. J. Santana, and R. H. C. Santana, ''Performance evaluation of resource management in cloud computing environments,'' *PLoS ONE*, vol. 10, no. 11, pp. 1–21, 2015, doi: 10.1371/journal.pone.0141914.

[36] C. Tang, S. Xiao, X. Wei, M. Hao, and W. Chen, ''Energy efficient and deadline satisfied task scheduling in mobile cloud computing,'' in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 198–205, doi: 10.1109/BigComp.2018.00037.

[37] D. Klusácek and H. Rudová, ''Performance and fairness for users in parallel job scheduling,'' in *Proc. Workshop Job Scheduling Strategies Parallel Process.*, 2012, pp. 235–252.

[38] R. Fagin and J. H. Williams, ''A fair carpool scheduling algorithm,'' *IBM J. Res. Develop.*, vol. 27, no. 2, pp. 133–139, Mar. 1983.

[39] M. Ajtai, J. Aspnes, M. Naor, Y. Rabani, L. J. Schulman, and O. Waarts, ''Fairness in scheduling,'' *J. Algorithms*, vol. 29, no. 2, pp. 306–357, Nov. 1998, doi: 10.1006/jagm.1998.0953.

[40] A. Demers, S. Keshav, and S. Shenker, ''Analysis and simulation of a fair queueing algorithm,'' in *Proc. Symp. Commun. Archit. Protocols*, 1989, pp. 1–12, doi: 10.1145/75247.75248.

[41] H. M. Chaskar and U. Madhow, ''Fair scheduling with tunable latency: A round-robin approach,'' *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 592–601, Aug. 2003.

[42] P. Goyal, H. M. Vin, and H. Cheng, ''Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks,'' *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 690–704, Oct. 1997.

[43] E. Hwang, J.-S. Kim, and Y. Choi, ''Achieving fairness-aware two-level scheduling for heterogeneous distributed systems,'' *IEEE Trans. Services Comput.*, vol. 14, no. 3, pp. 639–653, May 2021.

[44] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, ''Quincy: Fair scheduling for distributed computing clusters,'' in *Proc. ACM SIGOPS 22nd Symp. Operating Syst. Princ. (SOSP)*, 2009, pp. 261–276.

[45] R. Jain, A. Durresi, and G. Babic, ''Throughput fairness index: An explanation,'' *ATM Forum Contrib.*, vol. 99, no. 45, 1999.

[46] H. Zafar, Z. H. Abbas, G. Abbas, F. Muhammad, M. Tufail, and S. Kim, ''An effective fairness scheme for named data networking,'' *Electronics*, vol. 9, no. 5, p. 749, May 2020.

[47] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, ''Optimal tradeoff between sum-rate efficiency and Jain's fairness index in resource allocation,'' *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3496–3509, Jul. 2013.

• • •