Particle Swarm Optimization-Based Energy-Aware Task Scheduling Algorithm in Heterogeneous Cloud

Roshni Pradhan and Suresh Chandra Satapathy

Abstract Task scheduling in a cloud computing environment is one of the important aspects in the field of information technology. An efficient schedule is required to enhance the performance of the whole system which results a good quality of services (QoS). It is an NP-complete problem and attracts many researchers to use various meta-heuristics algorithms to develop task scheduling methods in the cloud environment. In most of the evolutionary methods, search space is large and initialized randomly which is one of the key components. In this paper, using the working mechanism of particle swarm optimization (PSO) algorithm, a set of solutions or schedules is created. Solution with efficient QoS parameters like makespan, cloud utilization, and energy consumption is chosen for allocation of the task into the heterogeneous multi-cloud environment. The algorithm undergoes a simulation process and is tested upon benchmark datasets which shows a better result in comparison to some existing cloud scheduling algorithms like min-min, max–min, cloud min-min scheduling (CMMS), cloud max–min scheduling (CMAXMS), and cloud normalized min-min max–min (CNXM) algorithms, genetic algorithm, etc.

Keywords Cloud · NP-complete · Makespan · Cloud utilization · Meta-heuristics

1 Introduction

The huge requirement of internet-based technology and application heads toward the rapid development of the cloud computing environment. It offers numerous services over the internet. Users of cloud computing avail facilities like computing resources, storage for data, system resources, etc. with an illusion of infinite computing facilities. All these are handled by data centers that are geographically distributed and situated in many regions. Cloud computing provides mainly three types of services

R. Pradhan (✉) · S. C. Satapathy
School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India
e-mail: roshni.pradhanfcs@kiit.ac.in

S. C. Satapathy
e-mail: suresh.satapathyfcs@kiit.ac.in

V. Bhateja et al. (eds.), *Communication, Software and Networks*, Lecture Notes in Networks and Systems 493, https://doi.org/10.1007/978-981-19-4990-6_40

infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [1]. Resource provision along with elastic computing is provided by the IaaS cloud. In PaaS, clients are given authority to use the cloud environment to build their software in the platform provided by cloud service providers (CSP). Users can avail of PaaS to use the software directly from the CSP. Virtualization is an important aspect of cloud computing, which enables a required number of virtual machines (VM) to the users [2]. It plays a vital role in task scheduling in a cloud computing environment. Different users submit their requests to the cloud environment in the form of a set of tasks that are allocated to machines or VMs in the cloud. Cloud computing follows the technique behind a combination of parallel and distributed computing. In a cloud system, the user aims to achieve efficient scheduling parameters after allocating all the tasks to the available resources [3, 4]. To provide an efficient and optimal schedule, it is required to analyze different optimization parameters which can be implemented in cloud task scheduling. In most of the research studies, evaluation parameters like makespan, cloud utilization, deadline time, energy consumption, etc. are considered [5–7].

Nowadays, energy consumption has become the crucial factor in cloud datacenters. To reduce it, cloud engineers are approaching nature-inspired optimized scheduling techniques which focus on less energy consumption in data centers.

Energy estimation and cloud resources usage are profoundly coupled [8, 9]. Low cloud resource usage is hollering an unsuitable measure of energy when they are completely used or adequately stacked. A new review on energy utilization and carbon emission of huge data centers is exceptionally high in the year of 2005, in the USA. Data centers in European locale have been assessed to devoured 1% sum of carbon emission, while in the USA, it is 2.8% around the same time [9–13]. In distributed computing, the basic equipment framework is hidden from the end client. Even though application solicitation can be thought about for low utilization of energy and high usage of cloud resources, cloud resources ought not to be over-burden or underloaded by the undertakings, rather ought to be utilized ideally [14].

In cloud task scheduling, different types of techniques are like heuristics scheduling, workflow scheduling, static scheduling, and dynamic scheduling [15]. According to the complexity of an algorithm, task scheduling can be described as heuristic, meta-heuristic, and hybrid task scheduling approaches [16]. Heuristics algorithms are mainly used to evaluate task scheduling algorithms like minimum execution time (MET), minimum completion time (MCT), shortest job to fastest processor (SJFP), min-min, max–min. In a cloud environment, generating a task schedule using a meta-heuristic algorithm is becoming the most approachable area of research. It deals with a multi-modal optimization problem. Task scheduling in the cloud is an NP-complete problem using meta-heuristics methods. Traditional meta-heuristics algorithms are particle swarm optimization (PSO), ant colony optimization (ACO), genetic algorithm (GA). In the recent study, a variety of approaches are found like Jaya algorithm, social-group optimization (SGO), teaching learning-based optimization (TLBO), etc. All these algorithms focus on a set of populations. In PSO, these populations are considered as particles. Scheduling is applied on this set of solutions to generate an efficient schedule. Using popular cloud scheduling parameters like makespan, it is evaluated [17].

In PSO, a set of particles is considered and initialized randomly. Each particle is assumed to be a task and is allocated to the available VM. Random initialization of particles is an important key factor in PSO. It leads to providing a wide range of search space for the particles which is a boost to an optimal solution [18]. It improves the performance of the PSO algorithm. The working mechanism of PSO is given through a flowchart in Fig. 1.

In this paper, the proposed method is to utilize heuristic scheduling algorithms to initialize the PSO search process. Makespan, cloud utilization, and energy consumption are also considered as optimization parameters [19]. The rest of the paper is listed as follows. Section 2 overviews the related works. Section 3 describes the system model of our scheduling approach. The scheduling algorithm is presented in

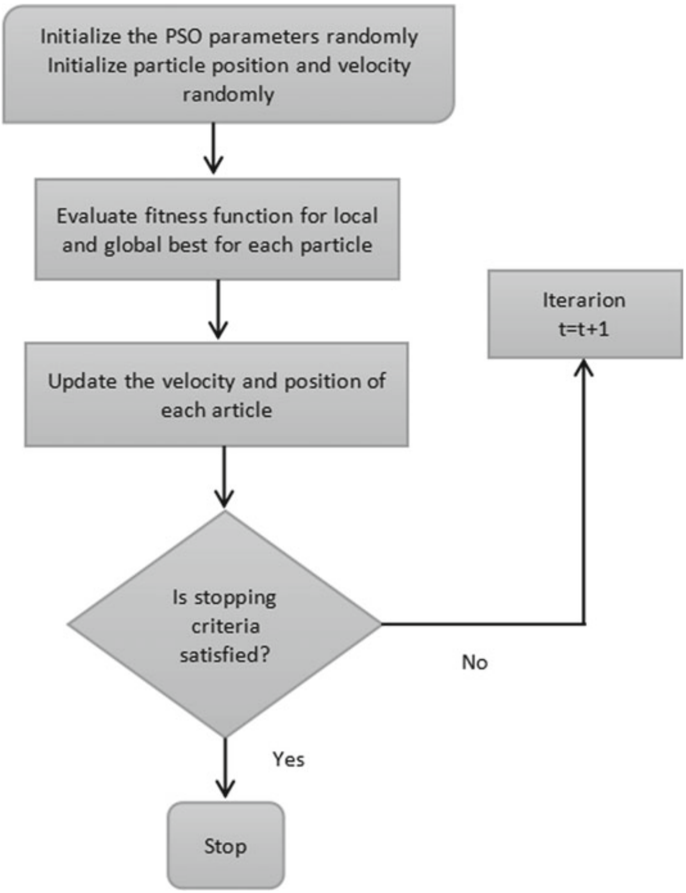

Fig. 1 PSO working mechanism

Sect. 4. Experimental evaluations and importance of dataset are described in Sect. 5. Section 6 concludes our work and highlights the future works.

2 Background and Related Work

In the current research area of interest, optimizing the scheduling parameters using an efficient schedule is gaining popularity. Besides this, resource allocation, cloud load balancing, security aspects in the cloud, energy efficiency are also major issues that attract attention. Clients or users, data centers, task managers are the components of this model. Data centers are responsible for the provision of an ample number of virtual machines required for the processing of requests. A task manager and scheduler are implemented to track the tasks before and after submitting into the resources and to prepare a schedule for upcoming tasks. Client requests in the form of tasks are updated in the task queue and later allowed for submission into resources.

In recent years of the survey, it has been noticed that many task scheduling algorithms are generated. These algorithms are either heuristics or meta-heuristics. Some of the algorithms focus on reducing makespan, whereas some algorithm designs a schedule that maximizes the cloud utilization values. There are some static traditional cloud scheduling algorithms proposed among which min-min, max–min, suffrage heuristics are included. The most important aspect is energy consumption in the cloud environment. In cloud computing, virtualization plays an important role, where data centers are engaged to create VMs. Data centers are globally distributed, and the number of data centers is growing rapidly to fulfill the client requests. In a recent study, it has been observed that 16% of data centers have been added. This results in a high increase in carbon emission, and the computation rate raised to 76%. To neutralize this impact, the USA has taken initiatives like the European code of conduct, the Energy Star program, and the 80 PLUS. The cloud environment has an architecture of two types, i.e., hardware and software. It is a tedious task to manage this architecture. The hardware environment can be manipulated by modifying the circuits to reduce energy consumption. In cloud data centers, the resource and energy consumption using an efficient task scheduling method is applied as suggested by [14]. But, the performance evaluation is not conducted in this paper. Energy-aware task consolidation is proposed by Refs. [19, 20] which aims to restrict CPU utilization. Hierarchical scheduling method is proposed to optimize the energy parameters in many research articles.

3 Models Used in the Proposed Algorithm

3.1 Cloud Task Scheduling Model

A cloud computing scenario is an architecture that incorporates several servers inside a data center. Servers are nothing but high-end processors and are treated as the resources for the client's requests. Requests are submitted in the form of a task queue. Tasks follow a specific algorithm, pass through a queue and task manager, and result the output. These algorithms are judged or analyzed in terms of theirs efficiency. Data transfer cost is also one among the quality of service (QoS) parameters which are treated as a vital factor. However, in most of the research papers, it is considered negligible. Using PSO, an efficient schedule is prepared, and along with that, QoS parameters like makespan, completion time, cloud utilization, and energy consumption are tracked.

3.2 Application Model

A group of tasks is taken in the form of application model. These are prepared to form a schedule, and a queue is generated. All the tasks are treated as independent and have their individual properties like execution time, completion time, etc. In this application model, execution time is used and previously estimated. It is stored in a task to machine/cloud matrix popularly known as the expected time to compute (ETC) matrix [2]. This application model preferably runs on a static multi-cloud environment. It is assumed to be no interference between the tasks and other I/O requests, cloud storage, and cloud resources.

3.3 Energy Model

A new energy model [21] is referred to estimate the total amount of energy consumption using the PSO scheduling method in a heterogeneous multi-cloud environment. To calculate this estimation, makespan and cloud utilization are formulated. It is found that average cloud utilization and energy consumption are linearly proportional according to a few research propositions. The execution time of the task on each cloud is referred to from the ETC matrix. Makespan is calculated as the completion time of the last task. Alternatively, it refers to the overall completion time required to execute all tasks in a multi-cloud environment. It is mathematically defined as given by Eq. (1).

$$M = \max\left(\sum_{i=1}^{n} \mathrm{ETC}(i,1) \times F(i,1), \sum_{i=1}^{n} \mathrm{ETC}(i,2) \times F(i,2), \ldots.p \sum_{i=1}^{n} \mathrm{ETC}(i,m) \times F(i,m)\right) \quad (1)$$

Energy consumption can be overly calculated as per Eq. (2). It is also seen that energy consumption is using overall cloud utilization for the calculation of energy used in a particular algorithm.

$$E_i = (P_{\max} - P_{\min}) \times U_i + P_{\min} \quad (2)$$

where

U_i = denotes the utilization of cloud i.

$P_{\max}$ = power use at the maximum load (or 100% cloud utilization).

$P_{\min}$ = minimum power consumption in the active mode (or as low as 1% utilization).

Here for idle resources, overhead to turnoff time is negligible. Hence, it is not considered.

Average energy consumption is denoted as in Eq. (3).

$$E = \frac{\left(\sum_{i=0}^{m} E_i\right)}{m} \quad (3)$$

where $1 \le i \le n$ and $1 \le j \le m$.

3.4 Scheduling Model in Cloud

In this scheduling model, a set of tasks and a set of machines are taken to compile the overall energy consumption using PSO algorithms. $T = \{T_1, T_2, T_3, \ldots\ldots.T_n\}$ is a set of independent tasks, and $C = \{C_1, C_2, C_3, \ldots.. C_m\}$ is a set of machines or clouds. Each task has some execution time on each machine. This mapping is represented in the expected time to compute (ETC) matrix. ETC matrix is given in Eq. (4), where Tth task execution time on Cth cloud is denoted.

$$\mathrm{ETC} = \begin{matrix} \mathrm{ETC}_{11} & \mathrm{ETC}_{12} & \cdots\cdots & \mathrm{ETC}_{1m} \\ \mathrm{ETC}_{21} & \mathrm{ETC}_{22} & \cdots\cdots & \mathrm{ETC}_{2m} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ \cdots & \cdots & \cdots\cdots & \cdots \\ \mathrm{ETC}_{n1} & \mathrm{ETC}_{n2} & \cdots\cdots & \mathrm{ETC}_{nm} \end{matrix} \quad (4)$$

4 Proposed PSO-Based Task Scheduling Algorithm

PSO is one of the traditional meta-heuristics scheduling algorithms, which provides the optimal solution for a group of particles. In task scheduling, PSO also outperforms to generate an efficient method. It is inspired by the social behavior of particles which later follows an evolutionary computational method. In this proposed task scheduling model, a group of swarm or particles is considered as a set of scheduled tasks. Each schedule has its behavior as the particles have. A predefined searched space is introduced to enhance the efficiency of the algorithm. Each particle represents a solution to the optimization problem, which is optimizing some cloud QoS parameters. Each particle is associated with position and velocity which helps them to move forward to the next step or position. Here, fitness function is evaluated at each step or position to identify the best solution or best particle. It will optimize the overall makespan, and the corresponding energy consumption will be calculated using the proposed algorithm. Velocity at the next position is defined by Eq. (5).

$$v_l^{t+1} = \omega v_l^t + c_1 r_1 \left(p_l^{\text{best}} - p_l^l\right) + c_2 r_2 \left(p^{\text{gbest}} - p_l^t\right) \tag{5}$$

where

p_1^t is the lth particle at iteration t.

v_l^t is the velocity of the lth particle at iteration t.

p^{best} is the best position found.

p^{gbest} is the best global position among all p^{best}

$1 < l < L$, where L is the population size.

Parameters c_1 and c_2 are the acceleration constants, r_1 and r_2 are random numbers between 0 and 1, and ω is the inertia factor.

In the proposed cloud task scheduling algorithm using PSO, solution (particle) which is denoted as S_1, S_2, $S_3 \ldots S_n$ is given in Fig. 1. For an instance, solution $S1$ is a schedule where T_1 is allocated to C_2, T_2 is allocated to C_2, etc. Optimization function (*F(x)*) is calculated for each solution after the end of each iteration, and p^{best} and p^{gbest} are chosen.

The parameters p^{best} and p^{gbest} are the velocities of the current particle and best particle. The best particle is one with less *F(x)* value (in of case minimization function). Here, schedule which gives less makespan is chosen. Hence, optimization parameter is a minimization function. A newly updated position of the particle can be calculated using Eq. (6).

$$p_{t+1} = (|v_{t+1}| + \text{mod}\, m) + 1 \tag{6}$$

where p_{t+1} = next position of particle (here, it is cloud/machine).

v_{t+1} = velocity at next position.

m = number of cloud or machine.

For each solution, the updated position for each machine is calculated. Here, it is assumed that the number of iteration is the same as the number of solutions or particles. An algorithm is given in the following table.

Algorithm: PSO based task scheduling
Input: 1. A set of n independent tasks 2. A set of m cloud 3. An ETC matrix **Output**: 1. Makespan 2. Energy consumption
1. Particle or solution = m^n is randomly initialized and a set of solutions is randomly chosen(40% of solution size). *iter*= 40% of m^n 2. $F(x)$ = makespan of selected solution is calculated . 3. for l= 1 to *iter* and t = 1 to *iter*(where iter= 40% of m^n) $v_l^{t+1} = \omega v_l^t + c_1 r_1 (p_l^{best} - p_l^l) + c_2 r_2 (p^{gbest} - p_l^t)$ For, m = number of cloud $p_{t+1} = (\lvert v_{t+1} \rvert + \mathrm{mod} m) + 1$ 4. Repeat step 3 until the termination condition is fulfilled. 5. Update fitness function value, choose the best schedule and calculate average energy consumption (Eqs. 2, 3)

4.1 *Illustration*

In Fig. 2, a set of solutions or particles is there which will undergo a series of steps to generate an optimized result. Best $F(x)$ value is treated as gbest, and the makespan of each solution is pbest. In the first iteration, the new velocity and position of the particle are calculated using Eqs. 5 and 6. Here, the position of the particle indicates, to which cloud the task will be assigned. For each solution, this process will be repeated, and simultaneously value of $F(x)$ is also calculated. After reaching the termination condition, the solution with the best $F(x)$ (minimization function) value is chosen. An example is illustrated in Fig. 3a–d.

In this example, the proposed algorithm is running for only one iteration. After a desired number of iteration, an updated solution will be obtained with a task to cloud allocation sequence. Makespan, cloud utilization, and energy consumption are

Fig. 2 Representation of a particle

Solution/ tasks	T_1	T_2	T_3	……….	T_n	$F(x)$
S_1	C_2	C_2	C_4	………	C_1	V_1
S_2	C_5	C_2	C_3	……….	C_2	V_2
S_3	C_7	C_1	C_6	……….	C_6	V_3
				……….		
S_n	C_1	C_7	C_3	……….	C_4	V_n

Task\cloud	C_1	C_2
T_1	150	18
T_2	32	7
T_3	20	3
T_4	50	15

(a) ETC matrix

Solution\Task	T_1	T_2	T_3	T_4	F(x)
S_1	2	1	1	1	102 (Pbest)
S_2	2	2	2	2	43
S_3	2	2	1	2	40 (Gbest)
S_4	2	2	1	1	70

(b) Initial solution set

Solution\Task	T1	T2	T3	T4	F(x)
S_1	1	1	1	1	252
S_2	2	1	1	2	52
S_3	1	2	2	2	150
S_4	1	2	2	1	200

(c) Updated position of cloud

Solution\Task	T_1	T_2	T_3	T_4	F(x)
S_2	2	1	1	2	52

(d) Optimized schedule after first iteration

Fig. 3 **a** ETC matrix. **b** Initial solution set. **c** Updated position of cloud. **d** Optimized schedule after first iteration

estimated for the solution. For the example given above, cloud utilization and the energy consumption are found to be 81% and 0.563 units, respectively.

5 Experimental Evaluation and Results

To evaluate the performance of different cloud scheduling parameters, both synthesized and benchmark datasets are taken. It is implemented using an Intel processor (2.6 GH) using. Benchmark datasets are heterogeneous. The 1024 × 32 and 512 × 16 datasets are represented in exam format for input to the proposed algorithm [22]. A geographical comparison of makespan, cloud utilization, and energy consumption for various cloud task algorithms like min-min, max–min, GA, etc. is given in Fig. 3 (1024 × 32) and Fig. 4 (512 × 16). Average cloud utilization and energy consumption are given in Table 1. In Table 1, instances of datasets are given in *u_x_yyzz* format, where

u = uniform distribution to generate these instances.
x = type of consistency (i.e., consistent ($\underline{C}$), inconsistent (I), or semi-consistent (S)).
yy = task heterogeneity (i.e., high (Hi) or low (Lw)).
zz = machine or cloud heterogeneity (i.e., high (Hi) or low (Lw)).

Note that these instances are especially used in cloud scheduling.

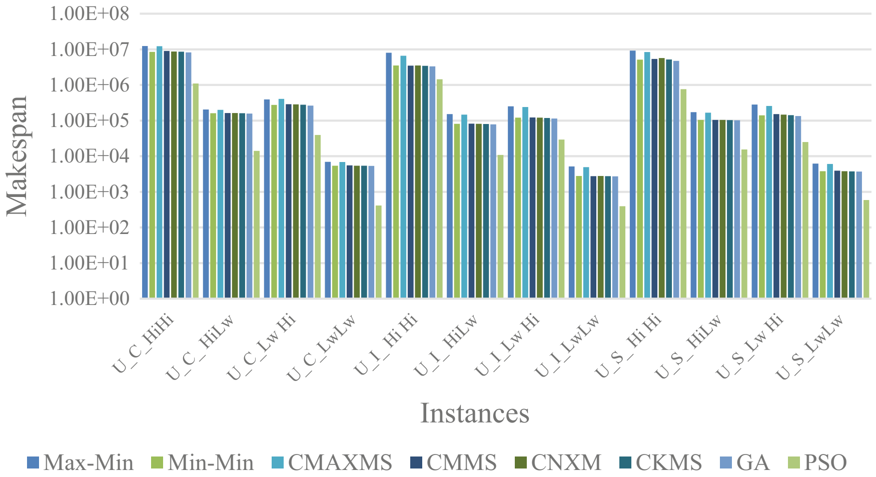

Fig. 4 Makespan comparison of 1024 × 32 benchmark dataset

Table 1 Cloud utilization and energy consumption for benchmark dataset

Instance	512 × 16 dataset		1024 × 32 dataset	
	Cloud utilization	Energy consumption	Cloud utilization	Energy consumption
U_C_HiHi	23.644943	22.36449	34.618919	23.461892
U_C_HiLw	49.596478	24.959648	45.784740	24.578474
U_C_LwHi	39.804482	23.980448	49.978962	24.997896
U_C_LwLw	42.455082	24.245508	58.015507	25.801551
U_I_HiHi	22.282265	22.228227	46.481632	24.648163
U_I_HiLw	31.728653	23.172865	43.350269	24.335027
U_I_LwHi	45.031673	24.503167	48.920189	24.892019
U_I_LwLw	48.759373	24.875937	51.085800	25.108580
U_S_HiHi	20.487280	22.048728	47.697536	24.769754
U_S_HiLw	53.183056	25.318306	59.066074	25.906608
U_S_LwHi	49.785385	24.978538	41.195431	24.119543
U_S_LwLw	53.189140	25.318914	52.242702	25.224270

6 Conclusion

In this paper, the working of the PSO algorithm in a heterogeneous multi-cloud environment has been portrayed. It is resulting from an efficient scheduling mechanism by taking a large search space. Hence, it is improving the overall makespan and generates cloud utilization and average energy consumption in cloud systems. These

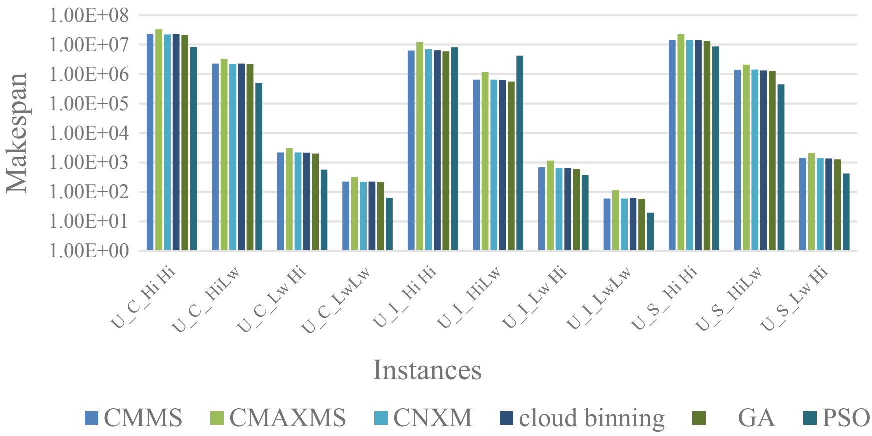

Fig. 5 Makespan comparison of 512 × 16 benchmark dataset

are cloud parameters required to evaluate the performance of the scheduling algorithm. PSO is one of the traditional evolutionary algorithms which performs better in a large search space. In the proposed algorithm, it is evaluating the overall energy consumption after completion of the scheduling method. It outperforms the existing algorithms in a distributed environment and evolutionary task scheduling. Synthesized and benchmark datasets are tested upon the proposed algorithm. The future work will be the implementation of the proposed algorithm in a real cloud environment and analysis of the proposed algorithm with advanced evolutionary scheduling strategies.

References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Futur Gener Comput Syst 25(6):599–616. https://doi.org/10.1016/j.future.2008.12.001
2. Li J, Qiu M, Ming Z, Quan G, Qin X, Gu Z (2012) Online optimization for scheduling preemptable tasks on IaaS cloud systems. J Parallel Distrib Comput 72(5):666–677. https://doi.org/10.1016/j.jpdc.2012.02.002
3. Panda SK, Pradhan R, Neha B, Sathua SK (2015) Fairness-aware task allocation for heterogeneous multi-cloud systems
4. Pradhan R, Dash AK (2019) A novel task scheduling algorithm in heterogeneous cloud environment using equi-depth binning method. Adv Wireless Technol Telecommun
5. Ibarra OH, Kim CE (1977) Heuristic algorithms for scheduling independent tasks on nonidentical processors. J ACM 24(2):280–289. https://doi.org/10.1145/322003.322011
6. Pradhan R, Panda SK, Sathua SK (2015) K-means min-min scheduling algorithm for heterogeneous grids or clouds. Int J Inf Process 9(4):89–99
7. Zhou X, Zhang G, Sun J, Zhou J, Wei T, Hu S (2019) Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. Futur Gener Comput Syst 93:278–289. https://doi.org/10.1016/j.future.2018.10.046

8. Kaabouch N, Hu W (2012). Energy-Aware Syst Netw Sustainab Initiatives. https://doi.org/10.4018/978-1-4666-1842-8
9. Koomey JG (2007) Estimating total power consumption by servers in the US and the world. Stanford University, Lawrence Berkeley National Laboratory
10. Barroso L, Hölzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
11. Bohrer PN, Elnozahy E, Keller T, Kistler M, Lefurgy C, McDowell C et al. (2002) The case for power management in web servers. Power-Aware Comput 261–289
12. Fan X, Weber W, Barroso L (2007) Power provisioning for a warehouse-sized computer. ACM SIGARCH Comput Archit News 35(2):13–23
13. Koomey J (2008) Worldwide electricity used in data centers. Environ Res Lett 3(3):034008
14. Meisner D, Gold B, Wenisch T (2009) PowerNap. ACM SIGARCH Comput Archit News 37(1):205–216
15. Pradhan R, Satapathy SC (2020) Task scheduling in heterogeneous cloud environment—A ICICC 2019. In: Advances in intelligent systems and computing, vol 1034. Springer, Singapore. https://doi.org/10.1007/978-981-15-1084-7_1
16. Mahmood A (2000) A hybrid genetic algorithm for task scheduling in multiprocessor real-time systems. http://www.ici.ro/ici/revista/sic2000-3/art05.html
17. Alsaidy SA, Abbood AD, Sahib MA (2020) Heuristic initialization of PSO task scheduling algorithm in cloud computing. J King Saud Univer—Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2020.11.002
18. Agarwal M, Srivastava GMS (2019) A PSO algorithm-based task scheduling in cloud computing. In: Ray K, Sharma T, Rawat S, Saini R, Bandyopadhyay A (eds) Soft computing: theories and applications. Advances in intelligent systems and computing, vol 742. Springer, Singapore. https://doi.org/10.1007/978-981-13-0589-4_27
19. Hsu CH, Chen SC, Lee CC, Chang HY, Lai KC, Li KC, Rong C (2011) Energy-aware task consolidation technique for cloud computing. In: 2011 IEEE third international conference on cloud computing technology and science. https://doi.org/10.1109/cloudcom.2011.25
20. Rizvandi NB, Taheri J, Zomaya AY, Lee YC (2010) Linear combinations of DVFS-enabled processor frequencies to modify the energy-aware scheduling algorithms. In: 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing. https://doi.org/10.1109/ccgrid.2010.38
21. Lee Y, Zomaya A (2010) Energy efficient utilization of resources in cloud computing systems. J Supercomput 60(2):268–280
22. Braun et al (2015) https://code.google.com/p/hcsp-chc/source/browse/trunk/AE/ProblemInstances/HCSP/Braun et al/uchihi.0?r=93. Accessed on 2 Feb 2015