**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

# A RESOURCE ALLOCATION SCHEME FOR CLOUD-BASED IOT APPLICATIONS BASED ON AN ENERGY-EFFICIENT MAKESPAN TASK SCHEDULING ALGORITHM

***Mamatha. T[1], Tressa Michael[2], Vithya Ganesan[3], Sujatha. S[4] , Arunabha Pal[5], Sumaiya Begum. A[6], Shital Kiran More [7], Saravana Selvam. N [8]**

[1]Professor, Department of Computer Science and Engineering, Atria Institute of Technology, Bengaluru, Karnataka, India, mamatha.t@atria.edu

[2]Assistant Professor, Department of Electronics and Communication Engineering, Rajagiri School of Engineering and Technology, Ernakulam, Kerala, India, tressa_m@rajagiritech.edu.in

[3]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Guntur District, Andhra Pradesh, India, vithyamtech@gmail.com

[4]Professor, Department of Electrical and Electronics Engineering, Sri Sairam College of Engineering, Anekal, Bengaluru, Karnataka, India, sujasakthi22@gmail.com

[5]Associate Professor, Centurion University of Technology and Management, Odisha, India, arunabha@cutm.ac.in

[6]Associate Professor, Department of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai, Tamilnadu, India, sumizahoor@gmail.com

[7]Assistant Professor, Department of Electronics and Telecommunications, Sinhgad Institute of Technology Lonavala, Pune, Maharashtra, India, shitalkmore4@gmail.com

[8] Professor, Department of Artificial Intelligence and Data Science, PSR Engineering College, Sivakasi, Tamilnadu, India, n.saravanaselvam@gmail.com

## Abstract

In cloud computing, task scheduling is the primary consideration when assigning resources dynamically to minimize makespan and increase performance. Due to the fact that data centers house a large number of computers, the rapid growth of cloud computing leads in massive energy consumption and emissions of carbon dioxide. Thus, in order to lower energy usage and carbon emissions, cloud service providers are searching for environmentally friendly solutions. Task scheduling has gained popularity as a result, where minimal energy usage and effective resource utilization are given careful thought. Although there are many applications and domains where cloud computing is widely employed, task and resource scheduling is still something that has to be improved. To lower costs, makespan, and energy consumption, this study presents the Cost and Energy-aware Task Scheduling Algorithm (CETSA). The suggested approach takes into account the load on each virtual machine as well as the trade-off between cost, energy consumption, and makespan in order to prevent virtual machines from overloading. The recommended method is put into practice with the help of the Cloud Sim toolkit and evaluated for various workloads. In addition, the recommended method lowers the rate of work failure, outperforms the prior algorithms in terms of energy

Received: August 10, 2025

consumption, and strikes a reasonable compromise between the makespan and the overall execution cost. Ultimately, the experimental findings demonstrate that the CETSA algorithm outperforms other algorithms in terms of energy usage.

**Keywords**:cloudcomputing; resource utilization; task scheduling;Makespan; energy consumption; Cost and Energy-aware Task Scheduling Algorithm (CETSA).

## INTRODUCTION

The concept of commercial technology application for public consumption is responsible for the growth of cloud computing as a leading technology. Because of its notable scalability, performance, low maintenance, and cost-effectiveness, cloud computing is known [1] for its capacity to dynamically assign resources on-demand. Task scheduling, which prioritizes minimal energy usage and effective resource utilization, has gained popularity as a result [2].Cloud computing is regarded as the foundation of the IT business, offering end users a dynamic virtualized resource pool [3].

The primary goals of cloud computing are to support businesses in meeting end-user needs and to offer an extremely effective platform for the optimal exploitation of computational features contained in organizations [4]. IoT devices are inherently limited by things like short battery life, poor processing power, and little storage capacity. To avoid these limitations, tasks requiring a lot of resources can be sent to more powerful cloud nodes[5].The optimal execution time is not always achieved when Internet of Things (IoT) processes are offloaded to the cloud for further processing, especially when there is resource conflict, under, over, or fragmentation[6].One of the key issues taken into account for effective resource management is task scheduling. Its goal is to distribute incoming jobs across the available processing power[7].

A direct connection to a server is not necessary when using the cloud's information technology delivery system, which uses web-based tools and apps to access resources from the Internet. In order to improve performance, cloud computing resources should be balanced [8].The primary intent of the cloud computing environment is to maximize the utilization of the available computer resources. Algorithms for scheduling are crucial to the optimization process. As a result, user tasks must be scheduled with an effective scheduling algorithm. Scheduling algorithms' primary goals are to reduce the total execution time while maximizing CPU utilization and allocating the load among the available processors. Organizing the tasks so they can be finished within the constraints particular to each scenario is the main objective of scheduling [9]. Increasing utilization and reducing makespan are the main goals of task scheduling in order to efficiently distribute work to the appropriate virtual machines (VMs)[10].

The primary contributions are described below.

1.An algorithm for energy-aware job scheduling is presented in this research. This technique may assign the parallel applications in workflows to the right processors within a specified deadline. It then handles the applications at the right times to minimize energy usage and achieve the necessary performance.

2. We verify the performance of CETSA by contrasting it with four widely recognized scheduling techniques. The amount of imbalance, makespan, cost, success rate, average waiting time, improvement ratio, and total energy consumption are the evaluation factors.

3.We examine the elements influencing our algorithm's performance.

The study has been organized as follows: Section II provides an explanation of the corresponding areas of task scheduling and cloud computing. Section III details on the proposed task scheduling framework. The experimental results are described in Section IV. The work is concluded in Section V with some recommendations for future research directions.

## RELATED WORK

As part of the infrastructure as a service (IaaS) idea, Xiaojin Ma et al. [11] developed a deadline and cost-aware scheduling algorithm that reduces the execution cost of a process under deadline limits. In order to accomplish task scheduling and resource allocation, Gawali et al. [12] proposed a heuristic approach that combines divide-and-conquer strategies, longest expected processing time preemption (LEPT), bandwidth aware divisible scheduling (BATS) + BAR optimization, and the modified analytic hierarchy process (MAHP). A semidynamic real-time task scheduling approach for bag-of-tasks applications in a cloud-fog environment was proposed by Abohamama et al. [13]. Task scheduling is formulated as a permutation-based optimization problem by the suggested scheduling algorithm. The Quickest Work Using datasets from Ahmed et al. [14], the First, Come, First Served (DVFS) scheduling algorithm and energy management algorithms (EMA) are compared.

Improved Artificial Rabbit Optimization based on Pattern Search (IARO-PS), as proposed by Paul et al. [15], is an improved version of Artificial Rabbit Optimization (ARO) that schedules dynamically independent requests (tasks) to overcome the challenges mentioned above. ARO's shortcomings are addressed by hybridizing the Pattern Search (PS) method to provide better exploration-exploitation balance. Alruwais et al. [16] introduced a Farmland Fertility Algorithm based Resource Scheduling for Makespan Optimization (FFARS-MSO) in a Cloud Computing Environment. Amer et al. [17] developed SMO_ACO, a novel hybrid multi-objective technique for tackling the scheduling problem.

An approach for energy-efficient workflow task scheduling with DVFS support was presented by Tang et al. [18]: DEWTS. A scheduling strategy known as the GA-FiFeS algorithm was presented by Hussain et al. [19] and links the genetic algorithm (GA) to the first feasible speed (FiFeS) technique. The GA-FiFeS algorithm guarantees fast reaction times while putting forward an energy-efficient schedule.Tang et al. [20] introduced the concurrent application job model and the design of heterogeneous computing systems. Subsequently, a model was developed to analyze the CPU-GPU utilization of the system computing node, and the energy usage during job execution was assessed. Wang et al. [21] set out to develop scheduling methods and demonstrate application experience for employing Dynamic Voltage Frequency Scaling (DVFS) technology in order to reduce the power consumption of parallel workloads in a cluster. An improved energy-efficient scheduling (EES) approach was put out by Huang et al. [22] in order to reduce energy usage and meet performance-based service level agreements (SLAs). Lee et al. [23] introduced a novel state transition technique for the problem of job scheduling in heterogeneous computing systems. It is a metaheuristic-based technique, called the Duplication-based State Transition (DST) approach. Wan et al. [24] introduced an energy-aware load balancing and scheduling (ELBS) system based on fog computing, with an emphasis on the intricate energy consumption challenges of manufacturing clusters.Yang et al. [25] examined an extensive analytical model intended to accurately evaluate the total

**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

energy efficiency (EE) in homogeneous fog networks, accounting for energy consumptions related to circuits, computation, and offloading.

## PROPOSED SYSTEM

The task scheduling problem is defined as the best way to allocate different jobs to a given number of virtual machines (VMs). It is feasible to assume that all submitting tasks are independent, that tasks cannot be moved across virtual machines (VMs), that VMs are heterogeneous, and that they differ in terms of processing power and efficiencybeing able to articulate the issue with task scheduling. Imagine a cloud system with m virtual machines (VMs) denoted by set V, V= {V1, V2,..., Vm}, where Vj denotes the VM that is located at position j in the cloud system. Furthermore, take into account n distinct tasks that users have submitted, which are denoted by set T, T= {T1, T2,..., Tn}, where Ti denotes the i-th task in the task queue.

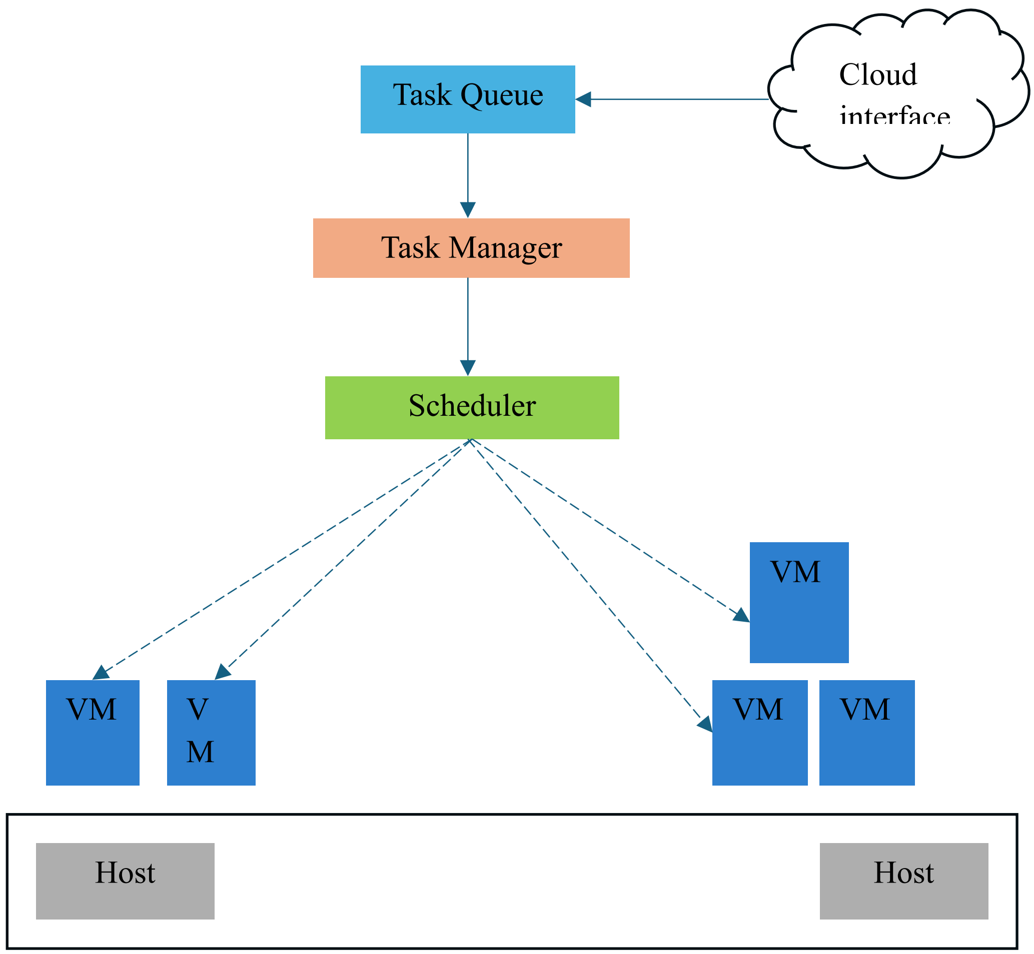
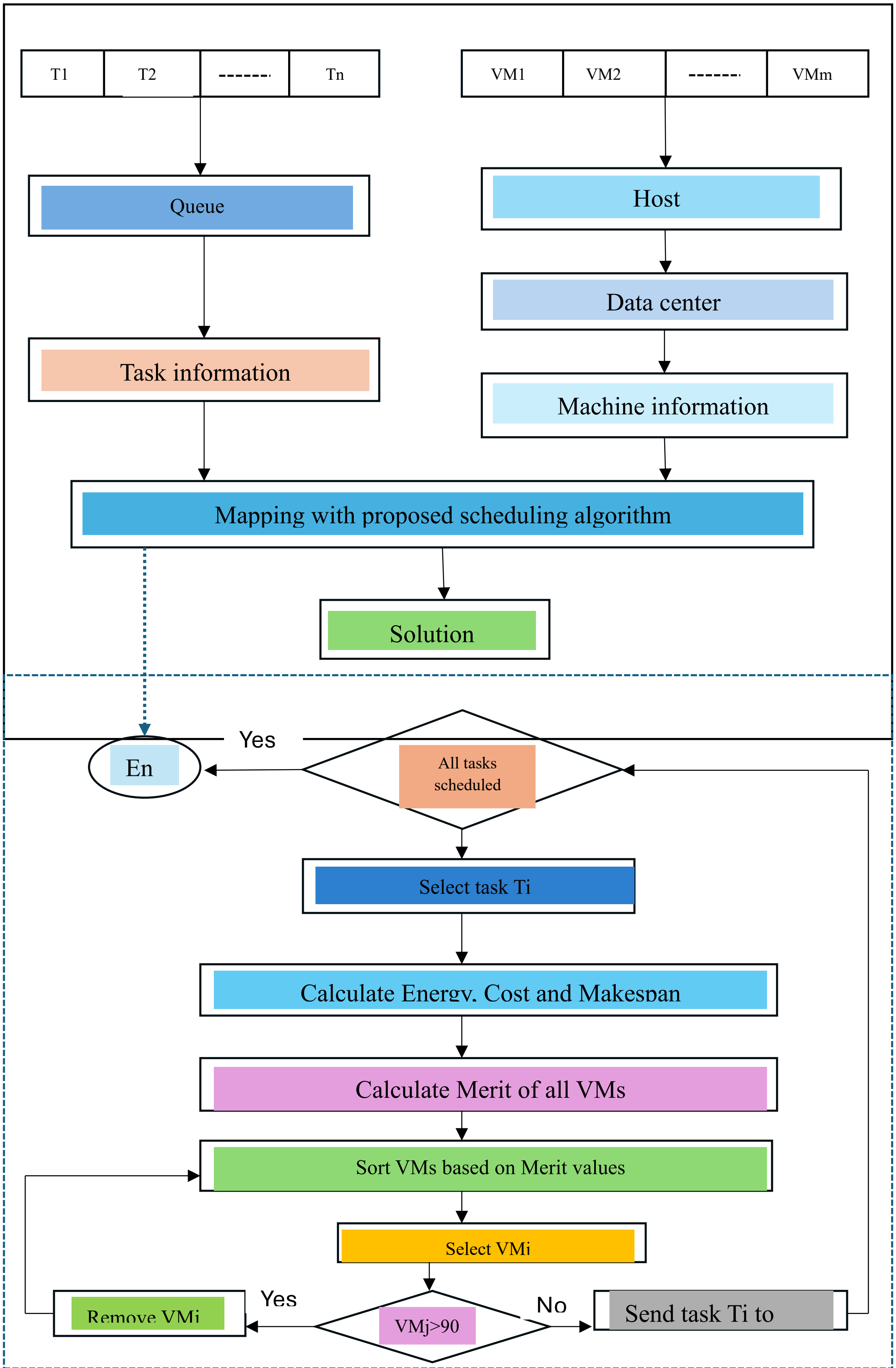



Fig.1

Cloud-Based Scheduling Process Framework

Received: August 10, 2025 

**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)




Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 5, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure.2Conceptual framework for task scheduling

### 3.1 Objective functions

The objective function's goal is to take performance metrics like makespan, cost, and energy usage into account. During the scheduling process, the suggested algorithm takes user happiness and cloud service provider goals (such energy consumption) into account. As a result, we design the objective function to minimize costs, makespan, and energy usage.

**3.1.1 Cost**:

The Execution Cost (EC), which refers to the whole cost of creating an application for a user, is usually the most quantifiable metric that is currently accessible. The customer wants to minimize schedule time as well as cost, so it is imperative to indicate the cost in terms of available resources. Equation (1) is utilized to calculate the Cost (C) associated with processing activities on Vj.

$$C_{V_j} = \sum_{j=1}^{m} sum\,(V_j) X \left(V_{PEj} X\; V_{ramj} X\; V_{bwj}\right) \tag{1}$$

where $V_{PEj}$,$V_{ramj}$, and $V_{bwj}$ indicate the cost of processing element, memory, and bandwidth performance of VMs, respectively.

**3.1.2 Makespan**

Makespan is the total amount of time required to complete each task in a work queue. An algorithm designed for intelligent scheduling aims to minimize the makespan. To improve our comprehension of makespan, we look at the words listed in Table 1. The definition of makespan is the maximum time allotted to do the ith task on the mth virtual machine, as illustrated in Fig. 2.

The makespan is the total amount of time needed for the resources to finish all tasksthat need to be executed and Equation (2) can be used to express it mathematically,

$$Makespan = \sum (finish(Ti)) - \sum \left(start(Ti)\right) \tag{2}$$

where the functions finish (Ti) and start (Ti)Give back the start time and finish time of the first and last jobs that were scheduled.

| Parameters | Definitions |
|---|---|
| **$t_i$** | $i^{th}$task |
| **$m_j$** | The virtual machine for mth |
| **$c_i$** | Time when task $t_i$appears |
| **$a_j$** | The duration of virtual machine $m_j$'s availability |
| **$e_{ij}$** | Time of Execution for $t_i$ on $m_j$ |
| **$c_{ij}$** | Time at which $t_i$ completed its execution |
| **makespan** | Maximum value of $c_{ij}$ |

Table.1 Parameters of makespan

Received: August 10, 2025 

**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

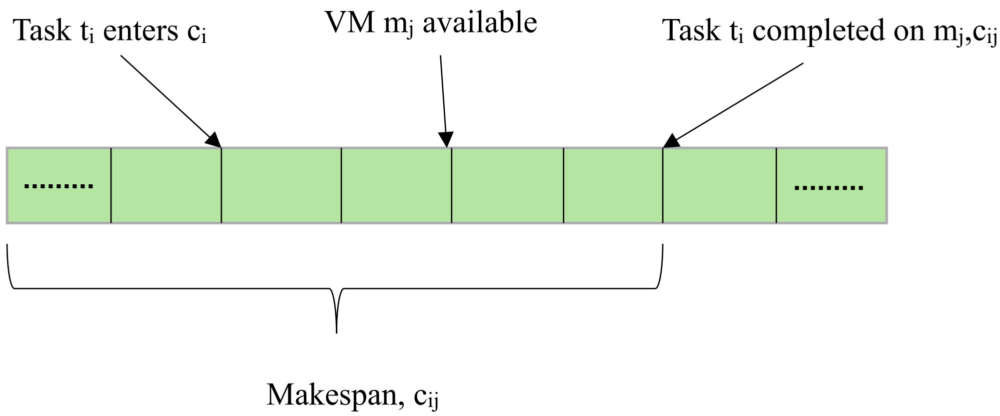



Figure.3Makespan and task lifetime

The Makespancan be calculated using the provided formula.

$$U_{ij} = \begin{cases} 1, & if\ T_i\ is\ assigned\ to\ V_j \\ 0, & if\ T_i\ is\ not\ assigned\ to\ V_j \end{cases} \tag{3}$$

$$E_j = \sum_{i=1}^{n} U_{ij}\ X\ TC_{ij} \tag{4}$$

Where TCij, which is found using Equation (5), represents the i-th (1 <i ≤ n) task completion time in the j-th (1 ≤ j ≤ m) VM.

$$\text{TCij} = \frac{L_i}{PE_j} \tag{5}$$

where $L_i$ denotes the task's duration in the i-th position.

3.1.3 Energy consumption:

Total energy consumption is the total quantity of energy utilized, both while the system is operating and when it is not. Equation 6 yields the energy consumption (EC) overall.

$$\text{EC} = \sum_{j=1}^{m} ([E_j\ X\ \alpha_j + (MS - E_j)\beta_j])\ X\ PE_j \tag{6}$$

The energy consumption of data centers is influenced by the use of CPUs, network interfaces, and storage devices. The CPU uses more energy than other system resources. A virtual machine's energy consumption can be separated into two categories: idle and active energy consumption. The two states of VM are taken

Received: August 10, 2025

into account when determining the total amount of energy used. While the amount of energy used in the active state is dependent on the virtual machine's processing speed, the amount used when a virtual machine is fully operating, its idle state makes up about 60% of it.

The Schema for CETSA One of the most crucial elements of cloud computing is work load balancing. A scenario where some nodes are overwhelmed while others are idle or have a small task to do is avoided by load balancing. Equations 7 and 8 must be used to determine each VM's load and capacity before determining whether or not load balancing is necessary.

$$V_{load} = {}^{(NL)}\!\big/_{V_{Mips}} \tag{7}$$

where N indicates the number of tasks assigned, L denotes the task's length, and VMips stands for the virtual machine's million instructions per second (MIPS).

$$V_{capacity} = PE_{Number} X \ PE_{Mips} \ X \ V_{bw} \tag{8}$$

where Vbw is the bandwidth connected to the virtual machine (VM), $PE_{Mips}$ is the MIPS capability of a processing element in the VM, and $PE_{Number}$is the number of processor units in the current virtual machine.Next, using Equation 9, the Processing Time (PT) of the present VM can be determined.

$$PT\_V_i = {}^{V_{load}}\!\big/_{V_{capacity}} \tag{9}$$

Finally, the standard deviation (σ) of the load is calculated according to Equation 10

$$\sigma = \sqrt{1\big/M \sum_{i=1}^{M} (X_i - X)^2} \tag{10}$$

where M is a VM property, Xi is the processing time of the active virtual machine, and X is the average processing time of the virtual machine.

Each VM's merit is displayed in Equation 11 according to its cost, makespan, and energy usage. The fraction denominator decreases as merit increases. Stated otherwise, a greater merit score corresponds to lower makespan, costs, and energy usage. Consequently, the suggested method looks for VMs with higher merit values in order to employ the VMs effectively. The formula that follows is utilized to ascertain each VM's merit based on cost, energy usage, and makespan:

$$Merit = \frac{1}{C + MS + EC} \tag{11}$$

### 3.2 CloudSim structure

The performance of the suggested task scheduling strategy is investigated based on the results of the simulation. The experiment with cloud computing was conducted using the CloudSim 3.0.3 simulator. The simulator can be used to compare the effectiveness of various work scheduling techniques. CloudSim is composed of three primary levels. The CloudSim core simulation engine, which is the third layer,

International Journal of Applied Mathematics
Volume 38 No. 5, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

models the queuing and communication between the components. The first layer, user code, provides configuration factors like the number of VMs, users, etc. The second layer controls the execution of key elements like cloudlets and resources when simulating.

| Parameters | Value |
|---|---|
| Number of data centers | 10-50 |
| Total number of VMs | 100 |
| MIPS of processing element | 500-4500 |
| Number of processing element per VM | 1-4 |
| VM memory (RAM) | 15-35 (GB) |
| Total number of tasks | 100-500 |
| Length of task | 100-2000 (MI) |
| Cost of storage | \$0.20 to \$0.30/GB |
| Cost of processing | \$1.25 to \$2.25/109 MI |
| Cost of transfer | \$0.25/GB |
| Load threshold | 90% |

Table.2 Simulation parameters

## RESULTS AND DISCUSSION

The CETSA algorithm's goal is to arrange the tasks so as to reduce expenses. Time and money are always at odds with one another. The algorithm that is being described takes the cost vs. time trade-off into account.

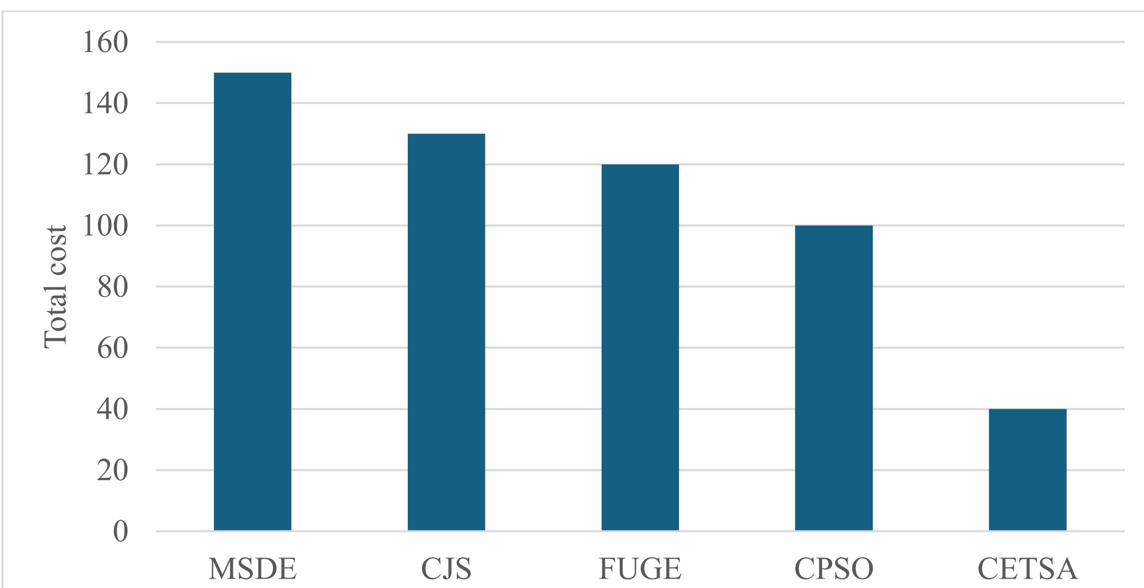



Figure.4 Total cost

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The combined cost of MSDE, CJS, FUGE, CPSO, and CETSA is displayed in Figure 4. It is noted that CETSA outperforms MSDE, CJS, FUGE, and CPSO in terms of cost, achieving 74%, 71%, 68%, and 58% reduced costs, respectively. As a result, CETSA generates more income and profit than other algorithms. This is a result of CETSA's efficient energy utilization and improved work scheduling. As a result, it lowers cloud providers' costs by improving energy efficiency.

Figure 5 shows the performance comparison in terms of energy usage; it should go without saying that as task counts increase, so does energy usage; in terms of energy conservation, the CETSA algorithm performs better and uses the least amount of energy overall.

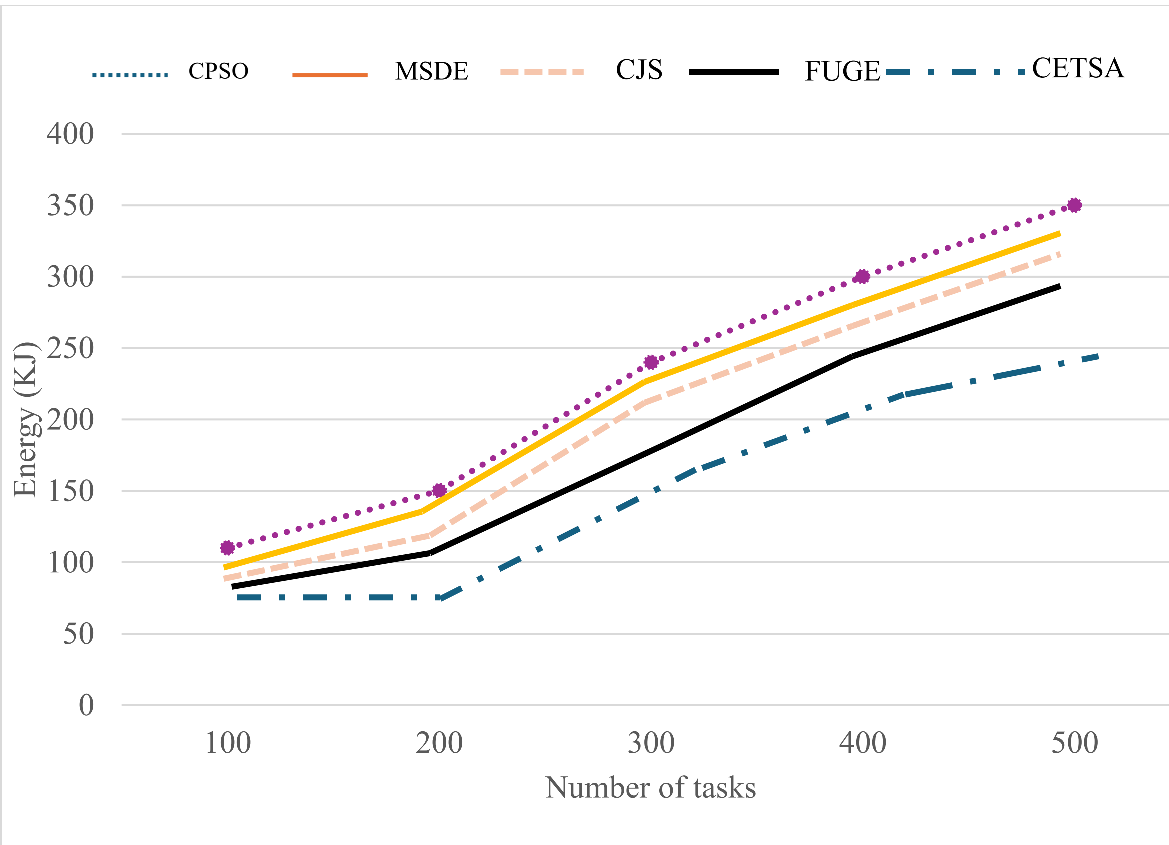



Figure.5Total amount of energy consumed

For jobs numbered 100 through 500, CETSA uses 2–20% less energy than FUGE does. Between 10 and 28% less energy is used by CETSA than by CJS for tasks of 100 to 500 executions, respectively.

The moment the last task is completed is known as the makespan or completion time. Determining the makespan is important since it aids in reducing energy usage and completing the task before the deadline.

The makespan comparison of the CETSA algorithm with alternative methods is displayed in Figure 5. Assume that there are a set number of data centers and that the quantity of work increases progressively from 100 to 500. The influence on makespan when increasing the number of tasks.

**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

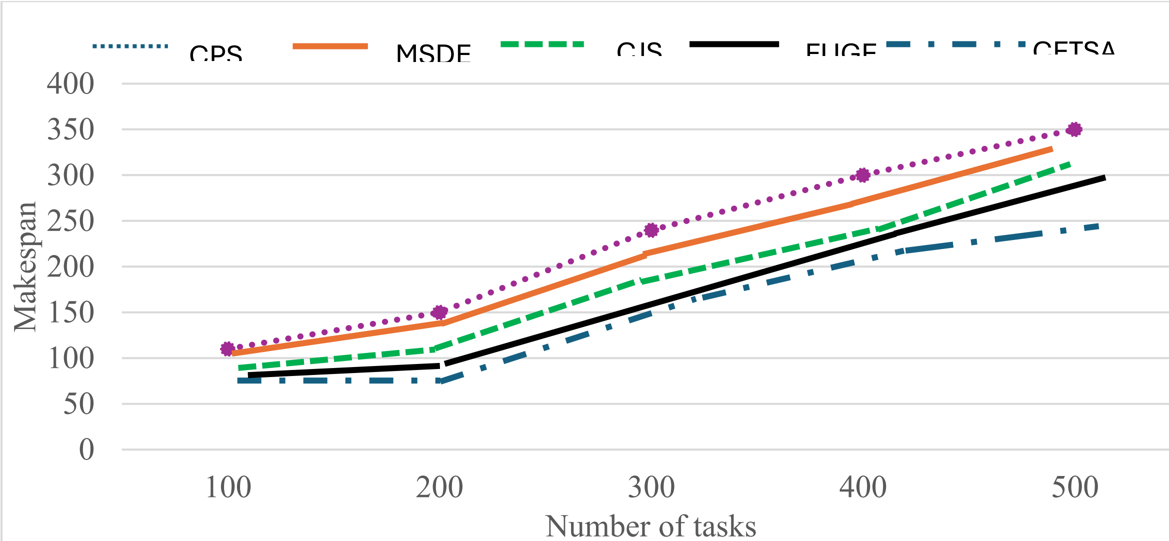



Figure.6 Makespan for the various task numbers

Figure 6 illustrates how the makespan grows along with the number of tasks. Compared to other algorithms, CETSA is more effective at minimizing the makespan. With 500 tasks assigned, it was found that the CETSA algorithm reduces the makespan by up to 26% when compared to CPSO, 23% when compared to MSDE, 14% when compared to CJS, and 9% when compared to FUGE.

The system's load is more evenly distributed and efficient when the degree of imbalance is modest. The degree of imbalance for each method when the number of jobs ranges from 100 to 500 is displayed. Figure 6 illustrates how the use of CETSA improves virtual machine load balancing performance. Because FUGE and CJS take task and machine capability into account while making scheduling decisions, they outperform MSDE and CPSO in limiting the degree of imbalance.

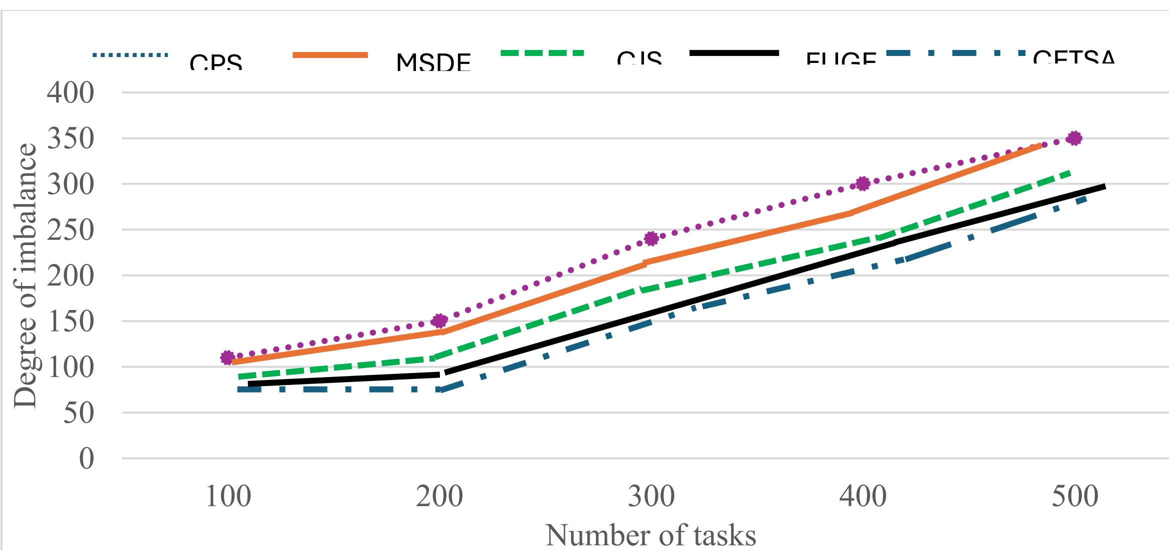



Figure 7. The degree of imbalance for varying task counts

Received: August 10, 2025

**International Journal of Applied Mathematics**
**Volume 38 No. 5, 2025**
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

| Reference | Algorithm | Performance measures | Deadline |
|---|---|---|---|
| Wan et al. [24] | Improved PSO | Workload ratio Robot mean difference | x |
| Yang et al. [25] | TDMA + MEETS | Energy efficiency | x |
| Proposed | CETSA | Cost, Makespan, Energy consumption | ✓ |

Table.3 Comparison Table

## CONCLUSION

In cloud computing, a thoughtful task scheduling algorithm is crucial as it minimizes makespan and optimizes resource usage while maintaining system stability.The task scheduling algorithm proposed in this paper takes energy and cost considerations into account in order to properly schedule jobs on cloud resources and maximize aspects like cost, energy usage, and makespan. Using the CloudSim simulator, the MSE, CPSO, CJS, and FUGE algorithms' performances were compared with that of the CETSA algorithm. After multiplying the number of virtual machines and running the chosen algorithms for each process, the optimal makespan optimization strategy may be found. The experiment's findings showed that, in compared to other algorithms, CETSA improves the success rate and improvement ratio more effectively while lowering costs, energy consumption, makespan, degree of imbalance, and average waiting time. In the future, other models might be utilized to more precisely characterize the patterns harmful traffic in the cloud computing environment uses. For improved cloud services, we will take into account additional Quality of Service (QoS) variables like availability, dependability, and SLA violation. Meta-heuristic algorithms will also be used to boost efficiency and produce better outcomes. In further study, the system reliability will be considered. A few particular settings would be taken into consideration to adjust the trials to the real scenario.

## Reference

1. Kaur, Rajbhupinder, Vijay Laxmi, and Balkrishan. "Performance evaluation of task scheduling algorithms in virtual cloud environment to minimize makespan." *International Journal of Information Technology* (2022): 1-15.
2. Panda, Sanjaya K., and Prasanta K. Jana. "An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems." *Cluster Computing* 22, no. 2 (2019): 509-527.
3. Mansouri, Najme, and Reyhane Ghafari. "Cost-Efficient Task Scheduling Algorithm for Reducing Energy Consumption and Makespan of Cloud Computing." Computer and Knowledge Engineering 5, no. 1 (2022): 1-12.
4. Al-Maytami, Belal Ali, Pingzhi Fan, Abir Hussain, Thar Baker, and Panos Liatsis. "A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing." *IEEE Access* 7 (2019): 160916-160926.

Received: August 10, 2025

5. Abohamama, Abdelaziz Said, Amir El-Ghamry, and Eslam Hamouda. "Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment." *Journal of Network and Systems Management* 30, no. 4 (2022): 54.
6. Al-Masri, Eyhab, Alireza Souri, Habiba Mohamed, Wenjun Yang, James Olmsted, and Olivera Kotevska. "Energy-efficient cooperative resource allocation and task scheduling for Internet of Things environments." *Internet of Things* 23 (2023): 100832.
7. Sharma, Mohan, and Ritu Garg. "An artificial neural network based approach for energy efficient task scheduling in cloud data centers." *Sustainable Computing: Informatics and Systems* 26 (2020): 100373.
8. Atiewi, Saleh, Salman Yussof, Mohd Ezanee, and MuderAlmiani. "A review energy-efficient task scheduling algorithms in cloud computing." In *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1-6. IEEE, 2016.
9. Jena, R. K. "Energy efficient task scheduling in cloud environment." *Energy Procedia* 141 (2017): 222-227.
10. Mangalampalli, Sudheer, Sangram Keshari Swain, and Vamsi Krishna Mangalampalli. "Prioritized energy efficient task scheduling algorithm in cloud computing using whale optimization algorithm." *Wireless Personal Communications* 126, no. 3 (2022): 2231-2247.
11. Ma, Xiaojin, Honghao Gao, Huahu Xu, and Minjie Bian. "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing." *EURASIP Journal on Wireless Communications and Networking* 2019, no. 1 (2019): 1-19.
12. Gawali, Mahendra Bhatu, and Subhash K. Shinde. "Task scheduling and resource allocation in cloud computing using a heuristic approach." *Journal of Cloud Computing* 7 (2018): 1-16.
13. Abohamama, Abdelaziz Said, Amir El-Ghamry, and Eslam Hamouda. "Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment." *Journal of Network and Systems Management* 30, no. 4 (2022): 54.
14. Ahmed, Adeel, Muhammad Adnan, Saima Abdullah, Israr Ahmad, Nazik Alturki, and Leila Jamel Menzli. "An Efficient Task Scheduling for Cloud Computing Platforms Using Energy Management Algorithm: A Comparative Analysis of Workflow Execution Time." *IEEE Access* (2024).
15. Paul, Santosh Kumar, Sunil Kumar Dhal, Santosh Kumar Majhi, Abhijeet Mahapatra, Pradosh Kumar Gantayat, and Swarupa Panda. "Optimizing Task Scheduling and Resource Utilization in Cloud environment: A novel approach combining Pattern Search with Artificial Rabbit Optimization." *IEEE Access* (2024).
16. Alruwais, Nuha, EatedalAlabdulkreem, Fadoua Kouki, Nojood O. Aljehane, Randa Allafi, Radwa Marzouk, Mohammed Assiri, and Amani A. Alneil. "Farmland fertility algorithm based resource scheduling for makespan optimization in cloud computing environment." *Ain Shams Engineering Journal* (2024): 102738.
17. Amer, Dina A., Gamal Attiya, and Ibrahim Ziedan. "An efficient multi-objective scheduling algorithm based on spider monkey and ant colony optimization in cloud computing." *Cluster Computing* 27, no. 2 (2024): 1799-1819.

18. Tang, Zhuo, Ling Qi, Zhenzhen Cheng, Kenli Li, Samee U. Khan, and Keqin Li. "An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment." *Journal of Grid Computing* 14 (2016): 55-74.
19. Hussain, Hameed, Muhammad Zakarya, Ahmad Ali, Ayaz Ali Khan, Mohammad Reza ChalakQazani, Mahmood Al-Bahri, and Muhammad Haleem. "Energy Efficient Real-time Tasks Scheduling on High Performance Edge-Computing Systems using Genetic Algorithm." *IEEE Access* (2024).
20. Tang, Xiaoyong, and Zhuojun Fu. "CPU–GPU utilization aware energy-efficient scheduling algorithm on heterogeneous computing systems." *IEEE Access* 8 (2020): 58948-58958.
21. Wang, Lizhe, Gregor Von Laszewski, Jay Dayal, and Fugang Wang. "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS." In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 368-377. IEEE, 2010.
22. Huang, Qingjia, Sen Su, Jian Li, Peng Xu, Kai Shuang, and Xiao Huang. "Enhanced energy-efficient scheduling for parallel applications in cloud." In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 781-786. IEEE, 2012.
23. Lee, Young-Choon, and Albert Zomaya. "A novel state transition method for metaheuristic-based scheduling in heterogeneous computing systems." *IEEE Transactions on Parallel and Distributed Systems* 19, no. 9 (2008): 1215-1223.
24. Wan, Jiafu, Baotong Chen, Shiyong Wang, Min Xia, Di Li, and Chengliang Liu. "Fog computing for energy-aware load balancing and scheduling in smart factory." IEEE Transactions on Industrial Informatics 14, no. 10 (2018): 4548-4556.
25. Yang, Yang, Kunlun Wang, Guowei Zhang, Xu Chen, Xiliang Luo, and Ming-Tuo Zhou. "MEETS: Maximal energy efficient task scheduling in homogeneous fog networks." *IEEE Internet of Things Journal* 5, no. 5 (2018): 4076-4087.

Received: August 10, 2025