Task Scheduling Algorithms In Cloud Computing: An Analysis

Niharika[1], Dr. Anita Ganpati[2]

Abstract- *Cloud Computing provides various services such as on-demand access, rapid elasticity, and broad network access that runs in a distributed network and is accessible through the internet. In the cloud environment, task scheduling plays an important role. The basic idea behind task scheduling is to improve the performance of cloud services by minimizing energy consumption and maximizing resource utilization. The aim of this paper is to study various task-scheduling algorithms and analyze them based on parameters like makespan, deadline, execution time, cost, and energy consumption.*

I. INTRODUCTION

Cloud computing is gaining a great scope among IT industries, academics, and individual users because of its various services like on-demand access to networks, pay-per-use, and resource pooling. Cloud computing is a technology that is used to hide the complexity of the internet or network access. The National Institute of Standards and Technology (NIST, 2014) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand - network access to a shared pool of configurable computing resources (e.g., Network, servers, services) that can rapidly be provisioned and released with minimal management efforts or service providers interaction" [1].

Cloud Computing services are categorized as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Deployment models of cloud computing are public, private, hybrid, and community [2].

Task scheduling and resource management are the two fundamental concerns with cloud computing. The cloud service providers should offer the necessary services, deploy virtual machines (VMs), and establish scheduling guidelines for assigning VMs to users' tasks [3].

1. Scheduling in Cloud Computing

Scheduling is the technique of allocating various tasks to available resources. Tasks are computational activities in the cloud, and each task requires different processing capabilities and different resource requirements such as CPU, memory, and number of nodes. Each task may have different parameters such as completion time (deadline), expected execution time, and job priority. A resource is something that is required by the task to carry out the operation e.g., a processor for data processing, or data storage devices.

2. Task Scheduling Algorithms in Cloud Computing

First Come First Serve [5]: In this algorithm, a queue of all incoming tasks is created and managed by the FCFS mechanism. Every new incoming task is placed at the end of the queue. The task in the queue is assigned to the currently available virtual machine (VM).

Shortest Job First [5]: In this algorithm, a queue is created based on the shortest job first. In this algorithm, the queue is in ascending order by placing the shortest tasks in front of the queue and longer tasks at the end of the queue.

Round Robin Algorithm [5]: The Round Robin Algorithm is a pre-emptive algorithm. This algorithm distributes the task queue in a round form or cyclic manner based on available VMs. This algorithm stores the tasks in the ring queue. Each job is given a quantum of time, if a task is not completed within turn it would be interrupted. Then the task is stored in the tailof the ring queue and waits for its next turn. This algorithm focuses on fairness among the scheduled tasks, tasks are executed in turn and never wait for the previous task to complete its execution.

Traditional Min-Min Heuristic Algorithm [6]: This method is based on the idea of Minimal Completion Time. The MinMin algorithm calculates the minimum completion time of each task across all machines and finds the shortest possible time for finishing each activity. Then assign the task to the machine that estimates minimal completion time. This process continues until all of the tasks are scheduled. Since tasks are scheduled on the fastest computers available, this algorithm's main benefit is a shorter makespan.

Traditional Max-Min Heuristic Algorithm [6]: This method is based on the idea of Maximum Completion Time. The Max-Min algorithm first finds the minimum execution time of all tasks. Then it chooses the task with the maximum execution time among all the tasks. The same procedure is repeated by Max-Min until all tasks are scheduled.

QoS Guided Min-Min Heuristic Algorithm [7] QoS-driven bandwidth needs of tasks that are taken into consideration by Min-Min. Tasks that require more bandwidths are scheduled before those that don't. Hence, if the bandwidth requirements for all tasks vary greatly, QoS- guided Min-min is preferable. This algorithm divides tasks into high-bandwidth and low-bandwidth categories. The tasks that demand a lot of bandwidth are scheduled first.

Resource Awareness Scheduling Algorithm (RASA) [8] The Resource Awareness Scheduling Algorithm is a hybrid algorithm. It combines the Max-Min algorithm and the Min-Min algorithm.

Genetic algorithms [11] Genetic algorithms are a type of optimization algorithm inspired by the process of natural selection. They work by creating a population of potential solutions and then using selection, crossover, and mutation operations to evolve the population toward better solutions**.** It is used to find the best schedule for executing a cloud workflow that meets the user's quality of service constraints.

Ant colony optimization (ACO) [12] Ant colony optimization (ACO) is a metaheuristic algorithm used to optimize the scheduling of scientific workflows in a local cloud scenario. ACO is inspired by the behavior of ants in finding the shortest path between their colony and food sources. In this algorithm, virtual machines (VMs) are considered food sources, and ants represent the workflow tasks.

Deadline-constrained Scheduling Algorithm (DCSA) [13] Deadline scheduling algorithms are a type of scheduling algorithm used in cloud computing environments to manage the execution of applications. These algorithms are designed to ensure that tasks are completed within a specified deadline while minimizing execution time and cost.

Deadline-Constrained Critical Paths Algorithm [14] DCCP algorithm first partitions the task graph into different levels based on their respective parallel and synchronization requirements. By using the average communication time and the shortest execution time, the algorithm determines which task node finishes the task at the earliest time. The maximum value of the earliest finish time for a task node on the same level is the sub-deadline for that node. When n task nodes and k server types are taken into consideration, the DCCP algorithm's complexity is O (n^2 k).

Deadline-Markov Decision Process (Deadline-MDP) Algorithm [15] The deadline-MDP algorithm divides the task graph into many independent branches and synchronization tasks. The overall deadline is divided into sub-deadlines for various tasks based on the minimum processing time required. Minimizing the execution costs of each branch task within the designated sub-deadline is the best course of action. Because each parallel branch job has a single sub-deadline and it runs on a different multi-task node with a longer execution path to satisfy its sub-deadline, this may increase the overall execution cost.

Driver of the Dynamic Essential Path (Deadline-DDEP) Algorithm [16] The deadline- constrained scheduling algorithm for cloud computing is based on the driver of the dynamic essential path (Deadline-DDEP) to solve the problem of deadline-constrained task scheduling in the cloud computing system. The algorithm adopts the dynamic sub-deadline strategy to solve the problem of the dynamic sub-deadline affected by the change of the dynamic essential path of each task node in the scheduling process. The proposed algorithm produces a remarkable performance improvement rate on the total execution cost that ranges between 10.3% and 30.8% under meeting the deadline constraint.

3. Task Scheduling Parameters [9]

Makespan

Makespan is defined as the amount of time taken to complete a set of tasks. For effective scheduling makespan of each task should be minimum.

4. Deadline

It describes the time by which a task should be finished. For effective scheduling algorithm constantly seeks to keep the work completed within the time frame.

5. Execution time

It describes the overall time taken by a task for its completion. Good scheduling algorithms have minimum execution time.

6. Energy consumption

The amount of energy consumed by the resources during the execution of the task.

7. Performance

8. It measures the scheduling algorithm's overall effectiveness in meeting the need of the user by delivering high-quality services.

9. **Quality of service (QoS)**

It includes satisfying execution time, cost, performance, and makespan.

10. **Load balancing**

It is a technique for distributing all the load in a cloud network over all nodes and links at once so that none of them are ever underloaded while others are always overloaded.

II. LITERATURE SURVEY

Hicham Ben Alla, et al. [30] proposed an efficient Deadline and Energy Aware Task Scheduling (DEATS) in Cloud Computing. The proposed algorithm increases the scheduling efficiency under deadline constraints and reduces the energy consumption of cloud resources. The proposed algorithm has been validated through the CloudSim simulator. The experimental results validated that the proposed algorithm effectively achieved good performance by minimizing the makespan, reducing energy consumption, and improving resource utilization while meeting deadline constraints.

Garima Gupta et al. [40] proposed two task scheduling algorithms out of which first is priority-based and the other is the earliest deadline first scheduling algorithm. The CloudSim toolkit has been used for the validation of proposed algorithm. The experimental results showed higher performance and improved memory utilization.

Maciej Malawski et al. [37] proposed the novel algorithms based on static and dynamic strategies for both task scheduling and resource provisioning. The proposed strategy has been validated through the CloudSim toolkit.

Zhongjin Li et al. [41] proposed a Cost and Energy-Aware Scheduling (CEAS) algorithm for cloud schedulers to minimize the execution cost of workflow and reduce energy consumption while meeting the deadline constraint. The proposed algorithm reduces energy consumption and has been validated through the CloudSim toolkit. The experimental result showedthat the CEAS algorithm is better in term of cost and energy-consumption.

Asraj Meena et al. [42] proposed Cost-Effective Genetic Algorithm (CEGA) with a deadline constraint. The proposed algorithm improved the virtual machines (VMs) performance variation and acquisition delay. The experimental result showed the highest hit rate for deadline constraints. This algorithm has a lower execution time than IC-PCP, RCT, and PSO and a lower execution cost than RTC, RCT, and PSO for deadly constraints.

Indukuri R. Krishnam Raju et al. [43] proposed Deadline Aware Two Stage Scheduling in cloud computing. The algorithm showed how to allocate resources in the form of virtual machines for the requested jobs. The experiment results showed that the algorithm performed well in terms of Average Waiting Time (AWT), Average Turnaround Time (ATT), Average Deadline Violation concerning Waiting time (ADVW), Average Deadline Violation concerning Response time (ADVR) as compared to FCFS, SJF,and Two Stage Scheduling Algorithms.

D. Komarasamy and V. Muthuswamy [27] proposed an Adaptive Multilevel Scheduling System (AMSS) algorithm. The proposed algorithm balances the load efficiently and bolsters resource utilization. CloudSim has been used for the validation of proposed algorithms. The experimental result showed that the performance of the AMSS algorithm is better than other algorithms.

Mohit Kumar and S.C. Sharma [24] proposed a dynamic scheduling algorithm for workload balancing among all VMs with elastic resource provisioning and de-provisioning based on the last optimal k-interval. The algorithm minimizes the makespan and increasesthe ratio of tasks to meet the deadline. The proposed algorithm has been validated through CloudSim. The experiment result showed that the proposed work improves the makespan and the number of tasks that meet the deadline.

S. C. Nayak and C. Tripathy [44] proposed AHP (Analytic Hierarchy Process) as a decision- maker in the backfilling algorithm. The proposed work minimizes the resource utilization ratio and the number of leased schedules. The experiment result proved that the performance of the backfilling algorithm is better by scheduling more leases and minimizing the lease rejection using AHP.

S. Abrishami and M. Naghibzadeh [19] proposed SaaS Cloud Partial Critical Path (SC-PCP) for workflow scheduling in SaaS Clouds, which minimizes the total execution cost while meeting a user-defined deadline. A Java-based simulator is developed to simulate the Cloud experiment. The experiment showed the computation time of this algorithm is very low for the decrease Cost and the Fair policies but is much longer for the Optimized policy.

Elias De Coninck et al. [28] proposed a Model-Driven Approach that investigates how to dynamically and automatically provision resources on the private and external clouds such that the number of workloads meeting their deadline is maximized. The proposed approach has been validated through CloudSim. The experiments showed that the approach completed most jobs before their deadline without using excessive resources.

Nazia Anwar and Huifang Deng [34] proposed Dynamic Scheduling of a Bag of Tasks based workflows (DSB), for scheduling scientific workflows to minimize the financial cost of leasing Virtual Machines (VMs) under a user-defined deadline constraint. The proposed algorithm minimized cost and maximized resource utilization. The experimental results validated that the proposed model produces better success rates to meet deadlines and financial costs.

Anurina Tarafdar et al. [38] proposed energy-efficient and makespan-aware scheduling algorithms for independent, deadline-sensitive tasks in the heterogeneous Cloud environment.

The proposed algorithm reduced the energy consumption of the large-scale virtualized Cloud infrastructure. The CloudSim Plus simulation platform has been for validation. The result showed a reduction of energy consumption, improved scheduling success, and achieved a proper trade-off between energy consumption and makespan of the tasks.

R.K. Jena [45] focused on task scheduling using Clonal Selection Algorithm (TSCSA) to optimize energy and processing time. The proposed approach is simulated by an open source cloudSim. The result was compared with the existing scheduling algorithm and the result shows that TSCSA provides an optimal balance.

A. Iranmanesh and H. R. Naji [46] proposed the use of new genetic operators as well as modified genetic operators. The algorithm optimized the scheduling workflow of tasks in the cloud environment. The approach has been validated through the CloudSim simulation platform. The experimental results showed a reduction in makespan and cost.

M. Kalra and S. Singh [47] proposed energy-aware scheduling of workflows in the cloud environment. The proposed approach focused on the hybrid approach for Energy-aware scheduling of deadline-constrained workflows (HAED) using the Intelligent Water Drops algorithm and Genetic Algorithm. The result showed that the solutions produced by HAED are of better quality in terms of accuracy and diversity than the non-dominated sorting genetic algorithm and hybrid particle swarm optimization.

K. Dubey and S.C.Sharma [48] proposed a novel hybrid task scheduling algorithm named Chemical Reaction Partial Swarm Optimization (CR-PSO). The algorithm schedules the multiple cloudlets with deadline constraints on hybrid cloud resources. The proposed algorithm has been validated through CloudSim. The result showed the that proposed algorithm enhanced the traditional CRO and standard PSO techniques and optimized the average execution time, makespan, energy, and cost parameters. The proposed scheduling algorithm is hybrid in nature and comprises two phases.

F. Ishikawa et al. [49] proposed a metaheuristic algorithm L-ACO as well as a simple heuristic ProLiS. The proposed algorithm optimized the execution time. The result showed that L-ACO obtained the highest success rate and achieved the lowest costs.

R.Ghafouri et al. [50] proposed a scheduling algorithm named CB-DT (Constrained Budget- Decreased Time). The proposed algorithm can decrease the makespan by satisfying the budget constraint of the workflow application. WorkflowSim based on CloudSim has been used for the experiments. The results showed that the proposed algorithm performs better than IC-Loss, BHEFT, and BDHEFT algorithms in most cases.

Z.G.Chen et al. [51] proposed ant colony system (ACS)-based approach for the resource scheduling problem of cloud computing under a cost-minimization and deadline-constrained model. The results showed that ACS can find better solutions at a lower cost than PSO and DOGA on various scheduling scales and deadline conditions.

M.A. Rodriguez and Buyya [27] proposed a resource provisioning and scheduling strategy for scientific workflows on Infrastructure as a Service (IaaS) clouds. The CloudSim simulation platform has been used for the validation of proposed algorihtm. The experimental results showed that the proposed algorithm has performed best as compared to the current state-of-the-art algorithms.

M. Zhang et al. [53] proposed an adaptive penalty function for the strict constraints compared with other genetic algorithms. The proposed approach has been validated through the workflow sim simulator platform. The results showed that the proposed algorithm performed better than the other state- of- the- art algorithms in the deadline- constraint meeting probability and the total execution cost.

K. Kalyan et al. [54] proposed Normalization-based Budget constraint Workflow Scheduling (NBWS) to address workflow scheduling under budget constraints in the cloud environment. The proposed approach has been validated through the CloudSim simulation platform. The results showed that NBWS performed better than other current state-of-the-art heuristics concerning budget constraints and minimizing the makespan.

III. RESEARCH METHODOLOGY

To conduct this research, analytical methodology is adopted in an appropriate manner to give a detailed analysis of task scheduling algorithms in cloud computing. Several relevant articles, studies, journals, papers and publications have been identified. High-quality and trustable reviewed journals and conferences like ACM, Springer, IEEE Xplore, and Science Direct are explored to get the relevant literature.

IV. ANALYSIS OF RESULTS

The main aim of the task scheduling algorithms in cloud computing is to improve performance, QoS, reduce overall cost and maintain efficiency among the tasks. Table I represents the task scheduling algorithm with their decision mode, scheduling parameters, response time, waiting time and scheduling type.

TABLE I: EXISTING SCHEDULING ALGORITHMS IN THE CLOUD COMPUTING

Scheduling Algorithm	**Decision Mode**	**Scheduling Parameters**	**Response Time**	**Waiting time**	**Scheduling type**
FCFS [5]	Non-pre-emptive	Arrival time	Low	High	Static
RR [5]	Pre-emptive	Arrival time, Time quantum	Can be high	High	Dynamic
MinMin [6]	Pre-emptive	Makespan, Expected completiontime	Moderate	Moderate	Dynamic
MaxMin [6]	Pre-emptive	Makespan, Expected completiontime	Moderate	Moderate	Dynamic
SJF [5]	Pre-emptive	Arrival time, Process time	Moderate	Low	Dynamic
RASA [8]	Pre-emptive	Makespan must be reduced	High	Low	Static

In Table I the task scheduling algorithms are evaluated based on various performance metrics such as response time, waiting time, and scheduling type. According to the analysis, the MaxMin algorithm performs well in terms of Makespan and expected completion time. The FCFS algorithm is non-preemptive, and the waiting time of this algorithm is high. In Table II represents the comparison of task scheduling parameters.

TABLE II: TASK SCHEDULING PARAMETERS IN CLOUD COMPUTING

References	**Makespan**	**Deadline**	**Energy Consumption**	**QoS**	**Load Balancing**	**Cost**
Ming Mao and Marty Humphrey [18]	×	×	×	×	×	╮

S.Abrishami and M. Naghibzadeh [19]	×	×	×	√	×	×
M. A. Elsoud et al. [20]	×	√	×	×	×	√
O. M. Elzeki et al. [21]	√	×	×	×	√	×
V. Vaithiyanathan et al. [22]	×	√	×	×	×	√
S. Bilgaiyan et al. [23]	×	×	×	√	×	√
M. Kumar and S. C. Sharma [24]	√	√	×	×	×	×
A. Negi et al. [25]	×	√	×	×	×	×
W. Song et al. [26]	×	√	√	×	×	√
D. Komarasamy and V. Muthuswamy [27]	×	×	×	×	√	×
E. De Coninck et al. [28]	×	√	×	×	×	×
N. Garg and M. S. Goraya [29]	×	×	√	×	×	×
H. Ben Alla et al. [30]	×	√	√	×	×	×
J. Meena et al. [31]	×	×	×	√	×	×
A. Gupta and R. Garg [32]	√	×	×	×	×	√
F. Juarez et al. [33]	√	×	×	√	×	×
N. Anwar and H. Deng [34]	×	√	√	×	×	×
P. Krishnadoss and P. Jacob [35]	√	×	×	×	×	√
Negar Chitgar et al. [36]	√	×	×	×	×	×
M. Malawski et al. [37]	×	√	×	×	×	√
A. Tarafdar et al. [38]	×	√	×	√	×	×
Indukuri R. Krishnam Raju et al. [39]	×	×	√	×	×	×

Table II compares the performance of various task scheduling algorithms based on different parameters like makespan, deadline, energy consumption and load balancing. The algorithms compared in the table are Min-Min, Max-Min, Genetic Algorithm, Ant Colony Optimization, and Particle Swarm Optimization. Concludes from Table II that two parameters deadline and makespan are mostly used. From Table II, the Ant ColonyOptimization algorithm performs the best in terms of makespan, deadline, and energy consumption. The Genetic Algorithm performs best in terms of deadline and energy consumption, while the Particle Swarm Optimization algorithm performs best in terms of cost.

V. CONCLUSION AND FUTURE SCOPE

Cloud computing provides various types of computing services like storage, databases, and software over the internet to offer flexible resources according to the needs of end users. Scheduling plays an important role in allocating resources to the users according to their demands. The paper analyzes the various task-scheduling algorithms with their parameters. After analyzing the different scheduling algorithms with parameters, it has been concluded that widely used parameters were deadline, energy consumption, makespan, QoS, load balancing and cost. Various metaheuristics algorithms were designed with these scheduling parameters. The Ant Colony Optimization algorithm performs the best in terms of makespan, deadline, and energy consumption. The Genetic Algorithm performs best in terms of deadline and energy consumption while the Particle Swarm Optimization algorithm performs best in terms of cost.

The future scope includes additional analysis and comparison of task scheduling algorithms in cloud computing, the creation of new algorithms, implementation and testing of proposed algorithms in real-world cloud environments, and investigation of the impact of taskscheduling algorithms on different types of cloud services.

REFERENCES

[1] M. N. Birje, P. S. Challagidad, R. H. Goudar, and M. T. Tapale, "Cloud computing review: concepts, technology, challenges and security," 2017. Available: https://www.researchgate.net/publication/304012876

[2] P. Sharma, S. Shilakari, U. Chourasia, P. Dixit, and A. Pandey, "A Survey On Various Types Of Task Scheduling Algorithm In Cloud Computing Environment," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, vol. 9, p. 1, 2020, [Online]. Available: www.ijstr.org

[3] E. Ibrahim, N. A. El-Bahnasawy, and F. A. Omara, "Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models," in *Proceedings - 2016 World Symposium on Computer Applications and Research, WSCAR 2016*, Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 65–71. doi: 10.1109/WSCAR.2016.20.

[4] R. Kaur and S. Kinger, "Analysis of Job Scheduling Algorithms in Cloud Computing," *International Journal of Computer Trends and Technology*, vol. 9, 2014.

[5] A. Agarwal, I. C. for I. T. University of Petroleum & Energy Study (Dehra Dūn, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, and Institute of Electrical and Electronics Engineers, *Proceedings on 2015 1st International Conference on Next Generation Computing Technologies (NGCT) : September 4th-5th, 2015, Center for Information Technology, University of Petroleum and Energy Studies, Dehradun*.

[6] X. He, X.-H. Sun, and G. Von Laszewski, "QoS Guided Min-Min Heuristic for Grid Task Scheduling *."

[7] Parsa, "RASA: A New Grid Task Scheduling Algorithm," *International Journal of Digital Content Technology and its Applications*, 2009, doi: 10.4156/jdcta.vol3.issue4.10.

[8] S. Arshad Ali, S. Member, and M. Alam, "A Relative Study of Task Scheduling Algorithms in Cloud Computing Environment."

[9] A. Patil and B. Thankachan, "Review On A Comparative Study Of Various Task Scheduling Algorithm In Cloud Computing Environment," 2020.

[10] A. Verma and S. Kaushal, "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud," *International Journal of Grid and Utility Computing*, vol. 5, no. 2, pp. 96–106, 2014, doi: 10.1504/IJGUC.2014.060199.

[11] A. Lal and C. Rama Krishna, "Critical path-based ant colony optimization for scientific workflow scheduling in cloud computing under deadline constraint," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2018, pp. 447–461. doi: 10.1007/978-981-10-7386-1_39.

[12] A. B. Kumar, K. P. Rani, and S. S. Kumar, "AN OVERVIEW OF CLOUD SCHEDULING ALGORITHMS," *Vidyabharati International Interdisciplinary Research Journal (Special Issue)*.

[13] V. Arabnejad, K. Bubendorfer, B. Ng, and K. Chard, "A Deadline Constrained Critical Path Heuristic for Cost-effectively Scheduling Workflows."

[14] "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids".

[15] X. Shao, Z. Xie, Y. Xin, and J. Yang, "A deadline constrained scheduling algorithm for cloud computing system based on the driver of dynamic essential path," *PLoS One*, vol. 14, no. 3, Mar. 2019, doi: 10.1371/journal.pone.0213234.

[16] Institute of Electrical and Electronics Engineers., Association for Computing Machinery., Sigarch., and IEEE Computer Society., *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for : 12-18 Nov. 2011 : [Washington State Convention Center, Seattle, WA : Connecting Communities through HPC]*. ACM, 2011.

[17] S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service Cloud," *Scientia Iranica*, vol. 19, no. 3, pp. 680–689, 2012, doi: 10.1016/j.scient.2011.11.047.

[18] IEEE Communications Society *et al.*, *International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013 22-25 Aug. 2013, Mysore, India ; [including symposia and workshops]*.

[19] O. M. Elzeki, M. Z. Reshad, and M. A. Elsoud, "Improved Max-Min Algorithm in Cloud Computing General Terms," 2012.

[20] V. Vaithiyanathan, R. A. Kumar, S. Vignesh, B. Thamotharan, and B. Karthikeyan, "An Efficient TPD Scheduling Algorithm for Cloud Environment."

[21] S. Bilgaiyan, S. Sagnika, and M. Das, "Workflow scheduling in cloud computing environment using Cat Swarm Optimization," in *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014*, IEEE Computer Society, 2014, pp. 680–685. doi: 10.1109/IAdCC.2014.6779406.

[22] M. Kumar and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," *Computers and Electrical Engineering*, vol. 69, pp. 395–411, Jul. 2018, doi: 10.1016/j.compeleceng.2017.11.018.

[23] A. Negi, K. V. K. Kishore, Annual IEEE Computer Conference, A. P. International Conference on Networks & Soft Computing 1 2014.08.19-20 Guntur, and A. P. ICNSC 1 2014.08.19-20 Guntur, *First International Conference on Networks & Soft Computing (ICNSC), 2014 19-20 Aug. 2014, Guntur, Andhra Pradesh, India ; proceedings*.

[24] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Trans Serv Comput*, vol. 11, no. 4, pp. 713–726, Jul. 2018, doi: 10.1109/TSC.2015.2466545.

[25] D. Komarasamy and V. Muthuswamy, "Deadline constrained adaptive multilevel scheduling system in cloud environment," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 4, pp. 1302–1320, 2015, doi: 10.3837/tiis.2015.04.002.

[26] E. De Coninck, T. Verbelen, B. Vankeirsbilck, S. Bohez, P. Simoens, and B. Dhoedt, "Dynamic auto-scaling and scheduling of deadline constrained service workloads on IaaS clouds," *Journal of Systems and Software*, vol. 118, pp. 101–114, Aug. 2016, doi: 10.1016/j.jss.2016.05.011.

[27] N. Garg and M. S. Goraya, "A Survey on Energy-Aware Scheduling Techniques in Cloud Computing Environment." [Online]. Available: https://sites.google.com/site/ijcsis/

[28] H. Ben Alla, S. Ben Alla, A. Touhafi, and A. Ezzati, "Deadline and Energy Aware Task Scheduling in Cloud Computing."

[29] J. Meena, M. Kumar, and M. Vardhan, "Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint," *IEEE Access*, vol. 4, pp. 5065–5082, 2016, doi: 10.1109/ACCESS.2016.2593903.

[30] A. Gupta and R. Garg, "Load Balancing Based Task Scheduling with ACO in Cloud Computing."

[31] F. Juarez, J. Ejarque, and R. M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 257–271, Jan. 2018, doi: 10.1016/j.future.2016.06.029.

[32] N. Anwar and H. Deng, "Elastic scheduling of scientificworkflows under deadline constraints in cloud computing environments," *Future Internet*, vol. 10, no. 1, Jan. 2018, doi: 10.3390/fi10010005.

[33] P. Krishnadoss and P. Jacob, "OCSA: Task scheduling algorithm in cloud computing environment," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 3, pp. 271–279, 2018, doi: 10.22266/ijies2018.0630.29.

[34] Dānishgāh-i Yazd and Institute of Electrical and Electronics Engineers, *ICEE 2019 : 27th Iranian Conference on Electrical Engineering : 30 April-2 May 2019, Yazd University, Yazd, Iran.*

[35] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Algorithms for Cost-and Deadline-Constrained Provisioning for Scientific Workflow Ensembles in IaaS Clouds," 2019. [Online]. Available: https://www.elsevier.com/open-access/userlicense/1.0/

[36] A. Tarafdar, M. Debnath, S. Khatua, and R. K. Das, "Energy and Makespan Aware Scheduling of Deadline Sensitive Tasks in the Cloud Environment," *J Grid Comput*, vol. 19, no. 2, Jun. 2021, doi: 10.1007/s10723-021-09548-0.

[37] R. TechCSE, "TASK SCHEDULING ALGORITHM TO OPTIMIZE CLOUD COMPUTING," *Int J Adv Res (Indore)*, [Online]. Available: www.garph.co.uk

[38] A. Negi, K. V. K. Kishore, Annual IEEE Computer Conference, A. P. International Conference on Networks & Soft Computing 1 2014.08.19-20 Guntur, and A. P. ICNSC1 2014.08.19-20 Guntur, *First International Conference on Networks & Soft Computing (ICNSC), 2014 19-20 Aug. 2014, Guntur, Andhra Pradesh, India ; proceedings*.

[39] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Trans Serv Comput*, vol. 11, no. 4, pp. 713–726, Jul. 2018, doi: 10.1109/TSC.2015.2466545.

[40] J. Meena, M. Kumar, and M. Vardhan, "Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint," *IEEE Access*, vol. 4, pp. 5065–5082, 2016, doi: 10.1109/ACCESS.2016.2593903.

[41] I. R. K. Raju, P. S. Varma, M. V. Rama Sundari, and G. Jose Moses, "Deadline aware two stage scheduling algorithm in cloud computing," *Indian J Sci Technol*, vol. 9, no. 4, pp. 1–10, 2016, doi: 10.17485/ijst/2016/v9i4/80553.

[42] S. C. Nayak and C. Tripathy, "Deadline sensitive lease scheduling in cloud computing environment using AHP," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 2, pp. 152–163, Apr. 2018, doi: 10.1016/j.jksuci.2016.05.003.

[43] R. K. Jena, “Energy Efficient Task Scheduling in Cloud Environment,” in *Energy Procedia*, Elsevier Ltd, 2017, pp. 222–227. doi: 10.1016/j.egypro.2017.11.096.

[44] A. Iranmanesh and H. R. Naji, “DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing,” *Cluster Comput*, vol. 24, no. 2, pp. 667–681, Jun. 2021, doi: 10.1007/s10586-020- 03145-8.

[45] M. Kalra and S. Singh, “Multi-objective Energy Aware Scheduling of Deadline Constrained Workflows in Clouds using Hybrid Approach,” *Wirel Pers Commun*, vol. 116, no. 3, pp. 1743–1764, Feb. 2021, doi: 10.1007/s11277-020-07759-4.

[46] K. Dubey and S. C. Sharma, “A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing,” *Sustainable Computing: Informatics and Systems*, vol. 32, Dec. 2021, doi: 10.1016/j.suscom.2021.100605.

[47] Q. Wu, F. Ishikawa, Q. Zhu, Y. Xia, and J. Wen, “Deadline-Constrained Cost Optimization Approaches for Workflow Scheduling in Clouds,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3401–3412, Dec. 2017, doi: 10.1109/TPDS.2017.2735400.

[48] R. Ghafouri, A. Movaghar, and M. Mohsenzadeh, “A budget constrained scheduling algorithm for executing workflow application in infrastructure as a service clouds,” *Peer Peer Netw Appl*, vol. 12, no. 1, pp. 241–268, Jan. 2019, doi: 10.1007/s12083-018-0662- 0.

[49] Z. G. Chen *et al.*, “Deadline constrained cloud computing resources scheduling through an ant colony system approach,” in *Proceedings - 2015 International Conference on Cloud Computing Research and Innovation, ICCCRI 2015*, Institute of Electrical and Electronics Engineers Inc., Feb. 2016, pp. 112–119. doi: 10.1109/ICCCRI.2015.14.

[50] M. A. Rodriguez and R. Buyya, “Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds,” *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 222–235, 2014, doi: 10.1109/TCC.2014.2314655.

[51] L. Liu, M. Zhang, R. Buyya, and Q. Fan, “Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing,” *Concurr Comput*, vol. 29, no. 5, Mar. 2017, doi: 10.1002/cpe.3942.

[52] K. Kalyan Chakravarthi, L. Shyamala, and V. Vaidehi, “Budget aware scheduling algorithm for workflow applications in IaaS clouds,” *Cluster Comput*, vol. 23, no. 4, pp. 3405–3419, Dec. 2020, doi: 10.1007/s10586-020-03095-1.