J. Electrical Systems 20-9s (2024): 1666-1676

[1]Suman Lata

[2]Dheerendra Singh

[3]Sukhpreet Singh

A Hybrid Approach for Cloud Load Balancing Optimization

Abstract: - In this research paper, a critical and novel approach is presented for cloud load balancing which delves into scheduling scientific workflows in cloud computing. These workflows are characterized by their complexity, demanding significant computational resources and sophisticated data processing capabilities. By leveraging a multi-objective genetic algorithm, this study strategically addresses the challenging task of efficiently distributing the workflows across cloud resources. This is particularly noteworthy as it involves a delicate balance of various conflicting parameters such as time, energy, cost, and adherence to quality of service (QoS) standards. The ingenuity of the presented approach is evident in the integration of an advanced ranking heuristic alongside the application of Bayesian methods for predicting the earliest finish time (PEFT). This dual strategy enhances the decision-making process in the allocation and migration of virtual machines (VMs), a cornerstone in cloud computing efficiency. This research goes beyond traditional methods by focusing on cost and time efficiency and integrating energy consumption considerations, an aspect increasingly relevant in today's environmentally conscious technological landscape. The results of this research, indicating substantial reductions in both cost and time delays, underscore the effectiveness of the proposed algorithm. By achieving these reductions, this approach offers a more sustainable and economically viable solution for cloud computing environments. Furthermore, the demonstrated potential of multi-objective genetic algorithms in this context opens new avenues for future research and development in cloud resource management and workflow scheduling.

Keywords: Cloud Computing, Hybridization, Load Balancing, Optimization, and Workflow Scheduling.

1 INTRODUCTION

A. Cloud Computing

The term" cloud" denotes a certain form of distributed and parallel system in which a group of virtualized devices is provisioned dynamically and displayed as a single or group of integrated computational resources based on the service level agreements (SLAs) negotiated among the service providers and the users. Cloud computing can be broken down into three distinct groups, each catering to a certain set of users with their own unique needs. We refer to them as" service models." The three most common service models are as follows: Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS) (Bello et al, 2021; Ibrahim et al, 2011). IaaS clouds, such as Amazon, offer virtualized storage and hardware to which subscribers can implement their services and applications. PaaS clouds, such as Microsoft Azure, offer an IDE for building and deploying cloud-based apps. There are two distinct Clouds that provide users with access to SaaS apps. The first category provides consumers with a fully functional application as a service that doesn't require any modifications or personalization on their part. Services like Google Docs and Calendar are just a couple of examples of the kinds of cloud-based productivity tools available. The second category, on-demand web services, provides users with basic web services that can be combined to create sophisticated software. Further, there are four primary categories of cloud hosts namely private cloud, public cloud, community cloud, and hybrid cloud (Javadi et al, 2012; Niu et al, 2017). A private cloud functions primarily for the use of a single company and is hosted either internally or by a third party. Whereas in public cloud the underlying infrastructure, applications, and data are made accessible to the public through a third-party service provider. These options are either entirely free (Ibrahim et al, 2011; Yahia et al, 2021) or have a small one-time fee. In the public cloud, companies like Google and Microsoft manage the underlying hardware as well as provide access to users exclusively through the Internet. Because of their massive density, public clouds often fail to complete jobs within the allotted time window. However, due to its limited computing resources, private clouds can't run all computation-intensive application tasks. When faced with this challenge, private companies are rushing to acquire public cloud services to boost their own data processing and storage capacities and public clouds would rather employ private cloud resources than set up their businesses. In a community cloud, multiple companies pool their resources to generate a single virtual server. Public sector organizations, financial institutions, etc., reap the most benefits from community clouds. However, hybrid clouds, which combine two or more cloud types (community, private, or public) are therefore an elegant solution in the current context because they improve upon previously mentioned qualities. As a result, public clouds are quickly implementing hybrid clouds, which seem to be capable of managing sensitive workloads generated by compute-

[1] Research Scholar, Chandigarh College of Engineering & Technical, Chandigarh sutharsuman2506@gmail.com
[2] dsingh@ccet.ac.in, Professor, Chandigarh College of Engineering & Technical, Chandigarh
[3] Assistant professor, Faculty of computing, Guru kashi university
Sutharsuman2506@gmail.com1, dsingh@ccet.ac.in2, sukhis005@gmail.com3

intensive apps by outsourcing some tasks to private clouds. To expand their existing applications, many companies today favour using a hybrid cloud (Niu et al, 2017). When scheduling tasks in a hybrid cloud environment, one of the most important considerations to make is the anticipated amount of time needed to complete the task. Scheduling time-sensitive processes in the hybrid cloud to minimize energy consumption and makespan within strict deadlines is a formidable challenge (Zhu et al, 2018; Sharif et al, 2014; Abazari et al, 2019).

B. Workflow Scheduling

Researchers have always been interested in workflow scheduling in cloud technology since it utilises most of distributed computing (Paknejad et al, 2021; Wu et al, 2018). Additionally, cloud computing is becoming necessary for the efficient execution of many workflow applications in massive computational systems like research and technology and the universal healthcare domain. Cloud systems typically consist of heterogeneous VMs sporting a wide range of computational resources with the limitless virtual capability of computational services, that can be utilized by workflow applications (Belgacem and Beghdad-Bey, 2020; Wu et al, 2014). But workflow scheduling is very challenging in a cloud environment because it requires allocating resources to tasks in a way that allows them to meet a set of performance standards. (Hussain et al, 2022; Malik et al, 2021; Yao et al, 2021; Ahmad et al, 2019; Han et al, 2021). A workflow is indeed a series of interconnected procedures, data and files that are sent among users in a workflow as per a predetermined set of rules, with the primary focus being on the automation of processes to reach a larger goal (Yu and Buyya, 2005; Adhikari and Nandi, 2019). Using workflows, programs can be structured as directed acyclic networks, where each node presents a task and edges indicate dependencies between tasks. Each activity in a workflow interacts with every other task in the pipeline through communication. Workflow management systems aid in making workflow execution easier. Through workflow scheduling, available resources are identified, and tasks are then assigned to those that are best suited to complete them. Workflow scheduling is a crucial component of workflow management, and its significance cannot be overlooked (Sardaraz and Tahir, 2019; Adhikari et al and Nandi, 2019; Ye et al, 2015; Chaudhary and Kalra, 2017) to make sure that workflows are scheduled effectively, algorithms for workflow scheduling are employed. Additionally, effective scheduling can greatly increase the efficiency of the workflow.

2 MOTIVATION

Research in workflow scheduling (Zhou et al, 2022; Tao et al, 2023) has been sparked by the need to assign computational resources to workflow tasks in the cloud. Workflow scheduling appears to be an NP-complete problem, making the development of a fast and efficient cloud-based optimum workflow scheduler challenging. Most of the existing literature treats cloud-workflow scheduling like a union bi-objective optimization problem, which overlooks the interests of either service providers or consumers. Consequently, developing a plan for the workflow programs is highly recommended. The methods used for scheduling workflows aim to maximize efficiency in certain areas of operation. Workflow scheduling in the cloud has emerged as a significant research area in tandem with the maturation of cloud computing technology and the widespread adoption of cloud platforms. The following factors make the issue complicated or hard: the difficulty of task-resource mapping in terms of NP-complete problem; a wide range of QoS constraints; the need to provide resources on demand; the presence of performance fluctuations and failures; the use of hybrid resources; and the need to optimize data storage and transmission. Thus, numerous research appeared in the literature, each concentrating on a certain feature. In the first part of this study, we investigate techniques for scheduling cloud-based workflows.

3 RELATED WORK

To foster a thorough understanding of the topic, this section reviews and summarises previous research, with a focus on approaches that address the workflow scheduling and cloud load balancing challenge in various contexts. Additionally, Table 1, presents a comparison of the current literature on cloud-based process scheduling.

Table 1: Examining recent efforts in the field of workflow scheduling and cloud load balancing

Year	Aim	Methodology	Features	Application Domain	Dataset	Limitations
Paknejad et al (2021)	In a cloud context, to carry out Chaotic enhanced PICEA-g-based Multi objective enhancement for	An improved coevolutionary algorithm with multiple objectives, named ch-PICEA-g	Candidate solutions a group of objective vectors	The method is based on the cost of execution, makespan, and energy usage	CloudSim simulator, WorkflowSim-1.0 toolkit	Premature convergence in already-in-use algorithms is a challenge that raises the quantity of iterations

	workflow scheduling					necessary to obtain an optimum solution.
Belgacem and Beghdad Bey 2022	To schedule multi-objective workflows in the cloud while balancing cost and makespan.	HEFT-ACO approach	Makespan and the cost	Scheduling workflow under resource allocation	Amazon EC2 cloud platform	The cloud computing model may experience significant issues with processing time and execution cost during allocation of resources, which could result in delays in the quality of service provided to consume
Hussain et al, 2022	To execute a multi-objective genetic algorithm inspired by quantum mechanics for scheduling workflow healthcare applications with soft and hard deadline restrictions in hybrid clouds.	A multi-objective technique has been adopted based on quantum mechanics	A quantumbit, A quantum gate	Healthcare processes utilizing quantumenabled GA and classical base algorithm	All standard methodologies are written in Python and run on a Windows server (Intel Core i7-9750H processor) operating at 8 GB and 2.60 GHz of RAM.	Consistency between energy consumption and makespan requires a reasonable balance.
Malik et al, 2022	Using queuing and thresholds, make a method for balancing the workload in cloud data centres in a way that saves energy	Particle Swarm Optimization (PSO) Workflow and Cloud Model	Modified Canopy Fuzzy C-means Algorithm (MCFCMA).	Queuing and threshold	Planet Lab datasets, Epigenomics datasets	Migration of virtual machines and adaptive thresholds
Zhou et al, 2023	Use of Firefly Optimizer's Makespan	Multi-Objective Normalizatio	Makespan, cost, and average	Scheduling problem Formulation	Strain green tensors (SGT) data	No conflicts

	and Security-Aware Workflow Scheduling for Cloud Service Cost Minimization.	n Workflow Scheduling	cloud utilization			
Tao et al, 2023	A Deadline-Budget-Restricted ACO for Cloud-Based Workflow Scheduling	An improved Firefly optimize	Makespan and security Cost-based minimizatio n	Cost-based minimization	Real-World Workflows data	The results show that the strategy can reduce financial costs by up to 54.0%, although it still has to be improved
Pillaredd y et al, 2023	Cloud-Based Multi-Objective Normalizatio n Workflow Scheduling	A Deadline, Budget Constrained Ant Colony optimization	Heuristic and metaheuristi c features	Execution Cost-based minimization	Real-World Workflows	When some resources' capacity exceeds users' needs and results in greater costs, users' budgets should be taken into account

4 PROPOSED SYSTEM

The proposed system's architecture is shown in Figure 1 and it implements the following steps for cloud load balancing and workflow scheduling.

Step 1. In the proposed approach, the workflow tasks are finished on different VMs Although the resources of various virtual machines (VMs) are usually fixed, how VMs are provisioned varies based on the tasks and workflows.

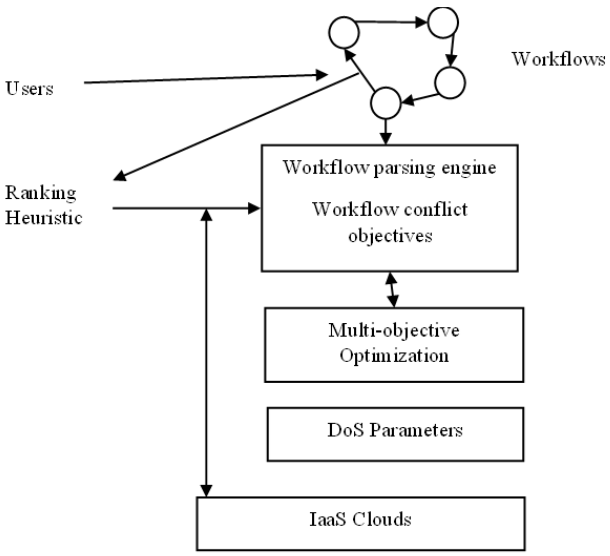

Figure 1: Proposed Approach Architecture

Step 2. Ranking: The proposed system uses the PEFT approach to provide task ranking. The selection of the PEFT heuristic for the ranking purpose in the proposed approach is due to the following reasons:

- The effective scheduling of tasks and optimized use of resources is the main goal of this study.
- Every optimization begins at random, which increases the time and sometimes does not result in convergence, reducing the approach's reliability.

Hence, PEFT is used to provide some initial input for problem space. The proposed methodology uses MPEFT due to the complexity of PEFT. It follows the following steps:

I. Extract the offspring set (S_i) containing all the direct and indirect successors tasks (t_i).

II. Offspring set (S_i) is calculated as:

$$S_i = \begin{pmatrix} \cup_{tj\ \varepsilon\ successor}\ offspring\ set\ tj \\ \phi\ otherwise \end{pmatrix} \tag{1}$$

Calculate DCT (Direct Calculative Time) representing minimum time required to perform the task t_i.

$$D\ (S_i) = w(\mathrm{ti}) + \sum_{tj\ \varepsilon\ successor\ ti} Xi, J \tag{2}$$

III. Combining the equations 1 and 2, results in ranking eq.3.

$R_{AP}\ (t_i)$= D(ti)

$$R_{AP}\ (t_i) = D(\mathrm{ti}) + \sum_{tj\ \varepsilon\ offspring\ set\ tj} Xij \tag{3}$$

Step 3. Optimization Modelling for Load Balancing using NSG-III

In this step, Load balancing is started by identifying uneven resources with the use of a metaheuristic technique. To optimize load balancing, the following steps were taken using NSG-III after the virtual machines (VMs) were mapped by the MPEFT algorithm in step 2.

1. User requests are defined by the workflow

W= $\{W_1, W_2, \ldots\ldots\ldots\ldots\ldots\ldots W_n\}$

2. The NSG-III in each solution is depicted in Figure 2.
3. **Initial population:** Population is a set of VMs and the ranking of tasks.

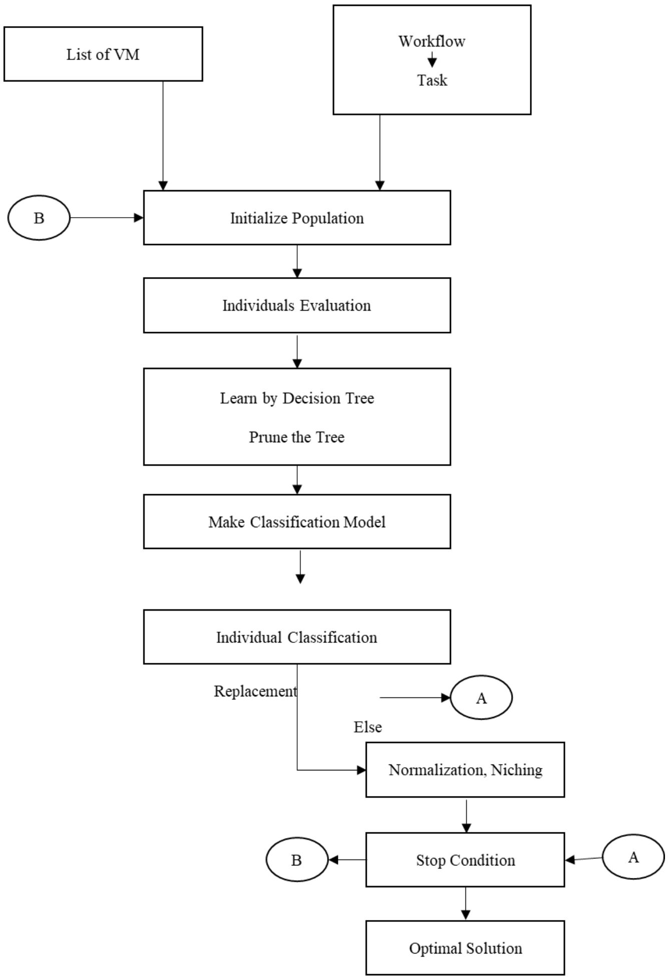

Figure 2: Optimization Modelling for cloud load balancing

4. Evaluation of individuals is done by a function called objective of the fitness function, $f(t_i)$ as:

$$f(t_i) = \sum_{i=1}^{N} \lambda 1 \, W.ET + \lambda 2 \, W.ET + \lambda 3 \, W.ET \quad (4)$$

Where, $f(t_i)$ fitness function, W priority matrix, and λ_1, λ_2, λ_3 learning parameter.
5. **Selection:** In this step choose N/2 population by uniform selection.
6. **Crossover:** The crossing of the two individuals is done by locating the parent crossing.
7. **Mutation:** Choose two VMs according to MPEFT ranking and swap the task of two VMs.
8. **Classification of Individuals:** In multi-objective optimization classify the individual according to their dominance. This process is done by DT and generalizes the model by pruning using eq. (4).
9. **Replacement:** Following the selection of the NSG-III operator, P_t is used to generate the new population X_t, which is then merged with P_t and N_t to form a new population of size 2N. The population is then classified using a decision tree to reduce the randomization effect.
Each of the objectives (Time, cost, and energy) added to the population
S_t, it should be $(S_t| > N)$
if $S_t = N$, NSG − III is applied to the starting population
10. Calculate the performance in deadline constraint
If task execution is within the deadline

$$ET(t_i, VM_n) = \frac{R(ti).W(ti)}{P\,((W(Mn)} \quad (5)$$

Unscheduled task

$$ST(t_i, VM_n) = \max(R\,(VM), Wi\,(ti\,, VM)) \quad (6)$$

Where: W_a waiting time, R Ready time

11. Final execution time of workflow

$$ET = \text{ST (ti , VMn)} + \text{ET (ti , VMn)} \quad (7)$$

Obj1 = $minimise\ (makespan)$

12. Budget constraints important for cost

$$C\ (\text{ti , VMn}) = FET\ (\text{ti , VMn}) * UC\ (VM) \quad (8)$$

Where: UC is Unit cost of VM

Obj2= $minimise\ (\sum_{i=1}^{N} C(ti))$

13. Dynamic voltage frequency scaling (DVFS)

$$Power_D = ACU^2 \quad (9)$$

Where: A is Switching activities, C is total capacitance, and U^2 is frequency

$$EC(t_i, VM_n) = Power_D . ET(t_i, VM_n) \quad (10)$$

14. **Obj3** = $minimise\ (\sum_{i=1}^{N-1} EC(t_i))$

5 EXPERIMENTAL RESULTS AND DISCUSSION

In this section we present the results attained from the comparative analysis of the proposed approach (PNSG3D) and different variants of multi-objective genetic algorithm based on Time span(sec), Cost ($), and Energy (Kw). To validate the suggested approach, we used a dynamic number of VMs as well as Ligo, and CyberShake workflows. Table 2,3, and 4 shows a comparison of the proposed work with various existing works, ABC, Tabu Search, GWO, and ACO, based on the makespan cost, and energy consumption. In Table 2, the number of virtual machines (VMs) used for task execution is represented by the rows, and the columns represent the various methodologies used for comparative analysis. The suggested work's performance was evaluated using three QoS metrics, time, cost, and energy consumption. The comparative analysis of proposed work with existing techniques using these performance metrics are shown in Fig 3, 4, and 5. The VMs used for task execution are represented on the X-axis, and the values of Time, Cost and Energy consumption acquired after simulation are represented on the Y-axis.

Table 1. Makespan (ms) analysis for load balancing optimization

	CyberShake					Ligo				
No of VMs	Proposed	ABC	Tabu Search	ACO	GWO	Proposed	ABC	Tabu Search	ACO	GWO
2	117.00	702.76	300.77	1334.58	942.48	318.59	624.21	302.00	857.67	1102.73
4	1153.36	1723.87	1784.19	1931.27	1751.48	1186.88	1586.30	1816.75	1454.71	1568.13
6	1277.94	1864.74	1918.78	2092.27	1904.81	1280.40	1704.01	1954.25	1601.00	1692.67
8	1444.44	2041.34	2090.85	2291.46	2079.02	1399.10	1844.84	2093.13	1788.04	1867.58
10	1609.28	2233.41	2267.19	2517.34	2291.77	1606.75	2037.21	2255.63	1979.71	2034.25
12	1781.95	2452.47	2516.35	2797.58	2542.41	1765.83	2249.40	2515.38	2250.54	2138.79
14	1972.79	2683.61	2768.47	3044.66	2838.19	1855.13	2400.74	2668.13	2524.58	2279.63
16	2265.18	2948.48	3107.91	3274.79	3082.76	2019.58	2675.21	3068.13	2893.54	2546.29
18	2421.31	3091.27	3257.61	3358.23	3258.23	2269.19	2908.03	3337.50	3056.25	2723.00
20	2617.44	3239.65	3458.86	3458.86	3358.86	2615.31	3245.16	3775.00	3275.00	3000.00

Table 2. Cost (Rs) analysis for load balancing optimization

	CyberShake					Ligo				
No of VMs	Proposed	ABC	Tabu Search	ACO	GWO	Proposed	ABC	Tabu Search	ACO	GWO
2	545.88	683.22	566.33	794.05	823.48	525.98	655.31	567.56	801.73	701.73
4	800.00	1322.08	1021.78	1552.06	1781.29	814.38	956.44	857.00	1126.00	1026.00
6	1769.17	2422.05	2082.34	2761.05	3011.22	816.00	1580.05	980.63	2278.79	2178.79

8	2533.42	3700.59	3439.77	4075.66	4101.43	1200.23	2584.81	1540.38	3754.25	3654.25
10	3265.73	4767.74	4711.55	5032.27	4869.08	4294.75	4560.38	4315.38	5136.63	5036.63
12	3758.38	5370.44	5352.98	5547.85	5356.63	5001.34	5566.26	5407.00	5833.83	5733.83
14	4065.83	5747.14	5742.72	5958.46	5679.35	5432.12	5910.60	5687.50	6257.46	6157.46
16	4403.41	6168.08	6174.20	6247.67	6232.12	6338.98	6502.85	6407.00	6726.46	6626.46
18	4545.08	6310.56	6321.15	6416.97	6345.23	6563.97	6723.00	6677.88	6886.19	6786.19
20	4696.57	6516.41	6512.80	6786.23	6453.12	6836.19	7015.34	7094.50	7094.50	6994.50

Table 3: Energy (Joule) analysis for load balancing optimization

	CyberShake					Ligo				
No of VMs	Proposed	ABC	Tabu Search	ACO	GWO	Proposed	ABC	Tabu Search	ACO	GWO
2	4.67	27.72	17.34	42.57	35.32	16.51	25.59	17.39	33.19	36.09
4	90.23	121.84	112.24	139.33	141.31	6.92	101.71	106.95	103.23	103.77
6	200.00	257.21	240.07	291.20	294.96	11.17	197.04	176.09	232.79	232.29
8	345.12	459.35	442.45	509.37	494.44	17.31	354.37	290.68	443.38	441.75
10	567.34	700.12	697.87	754.96	716.09	26.73	659.76	657.10	711.63	707.09
12	789.12	938.75	944.32	1001.45	947.89	306.49	937.88	950.69	970.13	944.72
14	980.23	1180.31	1191.57	1260.44	1192.46	309.88	1163.59	1169.79	1229.49	1181.19
16	1232.34	1458.65	1485.14	1523.59	1490.38	313.88	1468.49	1516.02	1539.20	1467.64
18	1456.12	1692.33	1724.18	1759.54	1728.62	289.27	1733.59	1802.77	1789.64	1711.65
20	1567.23	1951.21	1994.33	2049.02	1962.40	261.39	2052.10	2173.90	2073.90	1998.90

5.1 Comparison of results of proposed work with existing work

To show the efficiency of the proposed work, the simulation results were compared with several existing methodologies. Various approaches, including ABC, Tabu Search, GWO, and ACO were compared and displayed in graphical form where Figure 2,3, and 4 shows comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Time, Cost, and Energy consumption respectively using CyberShake workflow.

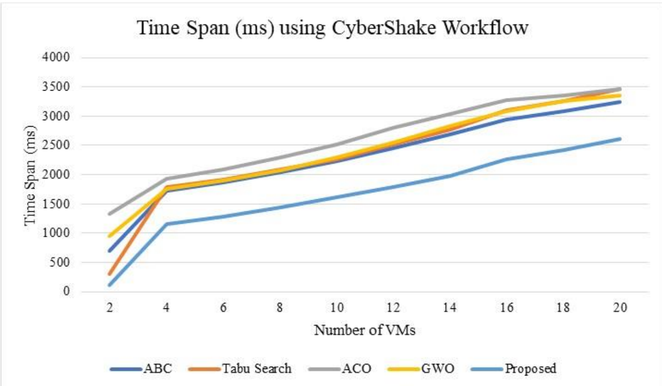

Figure 3. Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Time Span using CyberShake workflow.

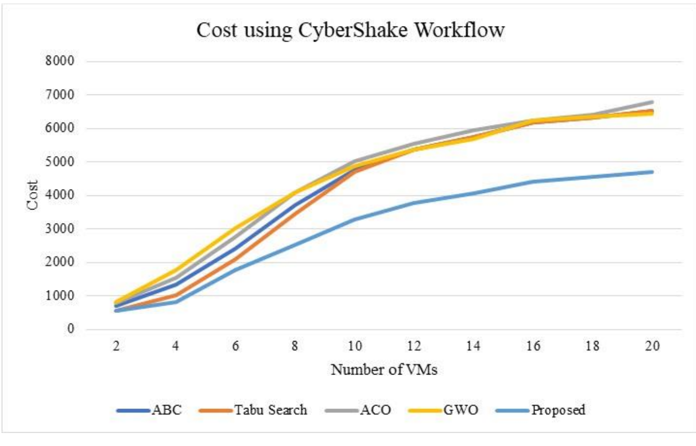

Figure 4.Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Cost using CyberShake workflow.

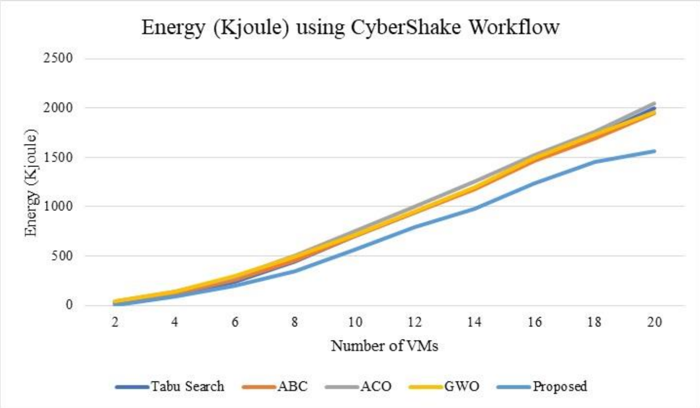

Figure 5. Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Energy Consumption using CyberShake workflow.

Furthermore, Figure 5,6, and 7 shows comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Time, Cost, and Energy consumption respectively using Ligo workflow

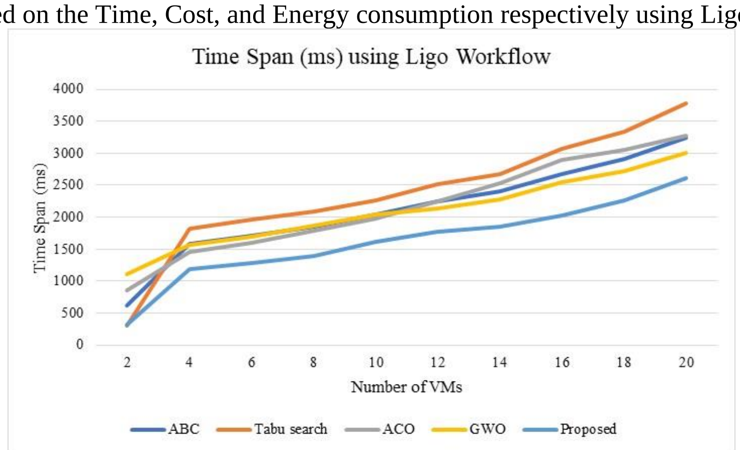

Figure 6. Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Time Span using Ligo workflow.

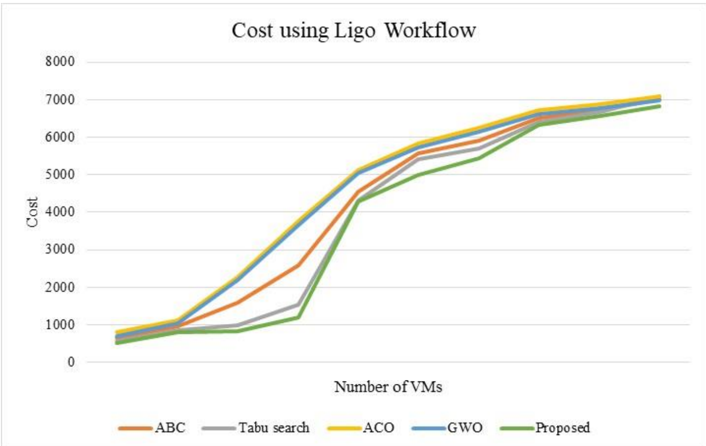

Figure 7. Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Cost using Ligo workflow.

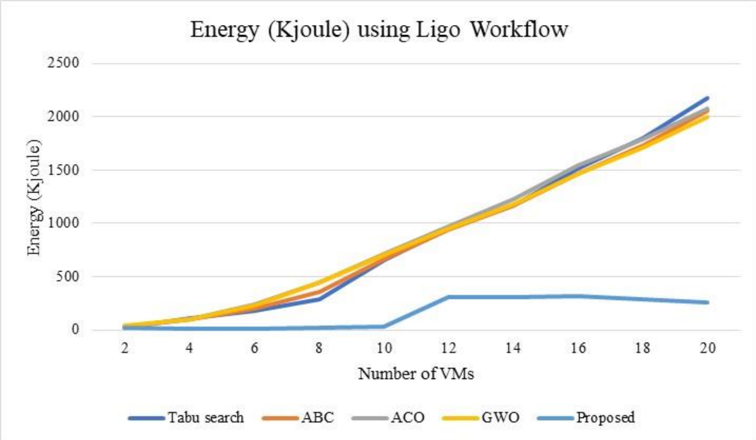

Figure 8. Comparative results of proposed work with existing ABC, Tabu Search, GWO, and ACO based on the Energy Consumption using Ligo workflow.

6 CONCLUSION

In the context of workflow load balancing in cloud computing, your research introduces a significant advancement by enhancing the conventional genetic algorithm through a hybrid, two-phase approach that optimizes the Pareto front using Bayesian optimization techniques. The first phase focuses on generating a list of solutions with low time complexity, achieved by optimizing the Pareto front through the Bayesian approach and ranking tasks based on the Predict Earliest Finish Time (PEFT). This phase effectively sets the groundwork for efficient scheduling. The second phase employs the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to address the premature convergence issue common in standard genetic algorithms and to efficiently handle conflicting objectives such as execution time and resource utilization. Our methodological rigor is further demonstrated through the use of CloudSim for simulating cloud environments and conducting extensive experiments with various workflows and virtual machines, each characterized by unique performance factors. The results of these experiments, showcase a reduction in costs by 5-6% and time delays by 8%, underlining the effectiveness of our approach (PNSG3D), in optimizing workflow load balancing in cloud environments. This innovative approach not only achieves optimal solutions but also significantly enhances the overall efficiency of cloud computing resource management.

6.1 Future Research Directions

Developing new multi-objective optimization algorithms that can handle the workflow scheduling problem efficiently and effectively more objectives and constraints, such as security and reliability, are integrating machine learning techniques to improve the performance of the algorithms. Overall, the proposed multi-objective optimization approach provides a promising framework for addressing the workflow scheduling problem in the cloud and has the potential to benefit various scientific and industrial applications.

REFERENCES

[1]. Abazari F, Analoui M, Takabi H, Fu S. MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm. Simul Model Pract Theory [Internet]. 2019;93(September 2018):119–32. Available from: https://doi.org/10.1016/j.simpat.2018.10.004
[2]. Adhikari M, Amgoth T, Srirama SN. A survey on scheduling strategies for workflows in cloud environment and emerging trends. ACM Comput Surv. 2019;52(4).
[3]. Adhikari M, Nandy S, Amgoth T. Meta heuristic-based task deployment mechanism for load balancing in IaaS cloud. J Netw Comput Appl. 2019;128(December):64–77.
[4]. Ahmad Z, Nazir B, Umer A. A fault-tolerant workflow management system with Quality-of-Service-aware scheduling for scientific workflows in cloud computing. Int J Commun Syst. 2021;34(1).
[5]. Belgacem A, Beghdad-Bey K, Nacer H. Dynamic Resource Allocation Method Based on Symbiotic Organism Search Algorithm in Cloud Computing. IEEE Trans Cloud Comput. 2022;10(3):1714–25.
[6]. Belgacem A, Beghdad-Bey K, Nacer H, Bouznad S. Efficient dynamic resource allocation method for cloud computing environment. Cluster Comput. 2020;23(4):2871–89.
[7]. Chaudhary N, Kalra M, Scholar PG. An improved Harmony Search algorithm with group technology model for scheduling workflows in cloud environment. 2017 4th IEEE Uttar Pradesh Sect Int Conf Electr Comput Electron UPCON 2017. 2017;2018-January:73–7.
[8]. Chopra N, Singh S. Deadline and cost based workflow scheduling in hybrid cloud. Proc 2013 Int Conf Adv Comput Commun Informatics, ICACCI 2013. 2013;840–6.
[9]. Feller E, Rilling L, Morin C. Energy-aware ant colony based workload placement in clouds. Proc - 2011 12th IEEE/ACM Int Conf Grid Comput Grid 2011. 2011;26–33.
[10]. Han P, Du C, Chen J, Ling F, Du X. Cost and makespan scheduling of workflows in clouds using list multiobjective optimization technique. J Syst Archit. 2021;112(April).

[11]. Han R, Guo L, Ghanem MM, Guo Y. Lightweight resource scaling for cloud applications. Proc - 12th IEEE/ACM Int Symp Clust Cloud Grid Comput CCGrid 2012. 2012;644–51.
[12]. Hassan MM, Song B, Han SM, Huh EN, Yoon C, Ryu W. Multi-objective optimization model for partner selection in a market-oriented dynamic collaborative cloud service platform. Proc - Int Conf Tools with Artif Intell ICTAI. 2009;637–44.
[13]. Ibrahim IM, Zeebaree SRM, Sadeeq MAM, Shukur HM. Task Scheduling Algorithms in Cloud Computing : A Review Hajar Maseeh Yasin Karwan Jacksi Zryan Najat Rashid Task Scheduling Algorithms in Cloud Computing : A Review. 2021;12(4):1041–53.
[14]. Ibrahim S, He B, Jin H. Towards pay-as-you-consume cloud computing. Proc - 2011 IEEE Int Conf Serv Comput SCC 2011. 2011;370–7.
[15]. Javadi B, Abawajy J, Buyya R. Failure-aware resource provisioning for hybrid Cloud infrastructure. J Parallel Distrib Comput [Internet]. 2012;72(10):1318–31. Available from: http://dx.doi.org/10.1016/j.jpdc.2012.06.012
[16]. Li J, Peng J, Lei Z, Zhang W. An energy-efficient scheduling approach based on private clouds. J Inf Comput Sci. 2011;8(4):716–24.
[17]. Malik N, Sardaraz M, Tahir M, Shah B, Ali G, Moreira F. Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds. Appl Sci. 2021;11(13).
[18]. Niu SH, Ong SK, Nee AYC. An improved Intelligent Water Drops algorithm for achieving optimal job-shop scheduling solutions. Int J Prod Res. 2012;50(15):4192–205.
[19]. Niu Y, Liu F, Fei X, Li B. Handling flash deals with soft guarantee in hybrid cloud. Proc - IEEE INFOCOM. 2017;
[20]. Paknejad P, Khorsand R, Ramezanpour M. Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment. Futur Gener Comput Syst [Internet]. 2021;117:12–28. Available from: https://doi.org/10.1016/j.future.2020.11.002
[21]. Pan Y, Wang S, Wu L, Xia Y, Zheng W, Pang S, et al. A Novel Approach to Scheduling Workflows Upon Cloud Resources with Fluctuating Performance. Mob Networks Appl. 2020;25(2):690–700.
[22]. Pandey S, Wu L, Guru SM, Buyya R. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. Proc - Int Conf Adv Inf Netw Appl AINA. 2010;400–7.
[23]. Pillareddy VR, Karri GR, Monws (????) Multi-Objective Normalization Workflow Scheduling for Cloud Computing. Applied Sciences 13(2):1101–1101
[24]. Sardaraz M, Tahir M. A hybrid algorithm for scheduling scientific workflows in cloud computing. IEEE Access. 2019;7:186137–46.
[25]. Sharif S, Taheri J, Zomaya AY, Nepal S. Online multiple workflow scheduling under privacy and deadline in hybrid cloud environment. Proc Int Conf Cloud Comput Technol Sci CloudCom. 2015;2015-February(February):455–62.
[26]. Wang X, Yeo CS, Buyya R, Su J. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. Futur Gener Comput Syst. 2011;27(8):1124–34.
[27]. Wu CM, Chang RS, Chan HY. A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. Futur Gener Comput Syst [Internet]. 2014;37:141–7. Available from: http://dx.doi.org/10.1016/j.future.2013.06.009
[28]. Wu T, Gu H, Zhou J, Wei T, Liu X, Chen M. Soft error-aware energy-efficient task scheduling for workflow applications in DVFS-enabled cloud. J Syst Archit [Internet]. 2018;84(November 2017):12–27. Available from: https://doi.org/10.1016/j.sysarc.2018.03.001
[29]. Wu Z, Liu X, Ni Z, Yuan D, Yang Y. A market-oriented hierarchical scheduling strategy in cloud workflow systems. J Supercomput. 2013;63(1):256–93.
[30]. Yahia HS, Zeebaree SRM, Sadeeq MAM, Salim NOM, Kak SF, AL-Zebari A, et al. Comprehensive Survey for Cloud Computing Based Nature-Inspired Algorithms Optimization Scheduling. Asian J Res Comput Sci. 2021;(May):1–16.
[31]. Yahia HS, Zeebaree SRM, Sadeeq MAM, Salim NOM, Kak SF, AL-Zebari A, et al. Comprehensive Survey for Cloud Computing Based Nature-Inspired Algorithms Optimization Scheduling. Asian J Res Comput Sci. 2021;(May):1–16.
[32]. Yu J, Buyya R. A taxonomy of workflow management systems for Grid computing. J Grid Comput. 2005;3(3–4):171–200.