International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Meta-Heuristic Technique Load Balancing for Cloud-Based Virtual Machines

Prof. Prabhakara B. K*[1], Dr. Chandrakant Naikodi[2], Dr. Suresh L[3]

Submitted: 25/10/2022 **Revised:** 09/12/2022 **Accepted:** 25/12/2022

Abstract*:* To efficiently develop cloud computing at the lowest price and the shortest time to deliver assets, task planning with Virtual Machines (VMs) has been crucial. the current report's several investigation holes for work scheduling optimization. This study included information and should be processed to solve the load balancing mechanism in the cloud environment. In this research, strategy-oriented combined support and load balancing structure has been created to maximize the use of virtual machines using similar weight distribution. The suggested method integrates heuristic and metaheuristic techniques to attain its optimal makespan& pricing efficiency HPOFT-MACO structure used two-step methodologies called Heuristics Predict Origin Finish Time (HPOFT) & Metaheuristic Ant Colony Optimization (MACO) to improve job management as well as cut costs and time.

Keywords: *Particle Swarm Optimization, Virtual Machine, Load Balancing, Heuristics Predict Origin Finish Time*

1. Introduction

Programs for which consumers have diversified, conflicting and variable quality of service (QoS) criteria are better suited to cloud technology. We use the internet to host apps, these features, operating models, or distribution patterns generate a hazy scenario because different applications have varying performance levels, work burdens, as well as demands for application programming scalability. Complicated implementation, installation, and delivery processes are set up on the Internet.A library to simulate Internet scenarios was called CloudSim.It offers basic categories to represent data centers, computing capabilities, virtual machines, programs, customers, and rules for administering various elements of the network, such as planning and delivery.

Management should be the process of hiding or distributing jobs in several establishments according to a limit or target criteria. Planning should be carried out in a way that maximizes or makes the best use of existing available to maximize or minimize the optimization problem and achieve the required level of service quality. Since there are currently no techniques that could provide an optimal result for scheduling issues throughout polynomial time, the objective is to highlight methods that could organize the jobs in less time, even if the answer wasn't optimum.

ACO, or Ant Colony Optimization, would be a kind of metaheuristics algorithm. The program was inspired by the ants in their search for food. From their colony, the ants move towards the food supply. The hypothesis would be that ants could identify the fastest route from their colony to a food supply using the routes traveled by ants in the past. In the process, ants emit pheromones. At first, ants choose courses arbitrarily. They also leave behind pheromones, however, they were weaker. Consequently, the shortest path finally has the strongest pheromone signal, which encourages all other ants to follow it. All ants eventually follow this path, which is also the fastest, after some time.

As load balancing techniques for cloud computing, manymetaheuristic programming methods created to improve job assignments have been explored. Experts have examined several scheduling strategies [1]. Research limitations have been identified through a previous literature review. ACO method balances the entire system and minimizes repetition [2]. The results are obtained by applying the fundamental ACO and FCFS approaches. Tasks were found to be digitally challenging, operationally interdependent, and non-repetitive [3].

The authors analyzed a variety of inputs, including independent operations without significant or budgetary constraints.ACO settings were silenced at random [4-6]. The min-min programming method was improved through load balancing, resulting in an increase in resource consumption and a decrease of a moment. In

[1]Department of ISE, A J Institute of Engineering &Technology, Mangaluru, VTU Belagavi, India.ORCID ID: 0000-0001-6001-7259*
[2]Department of Studies and Research in Computer Science (PG), Davangere University, India.
[3]Department of ISE, RNS Institute of Technology, Bengaluru, VTU Belagavi, India
** Corresponding Author Email: prabhakararesearch@gmail.com*

addition, operating schedule and budget limits have been prioritized throughout their research [7]. Researchers also compared the lifespans of ACO-based internet activity planning of FCFS and Round Robin methods, and they concluded that ACO outlasted the two equivalent approaches [8]. However, constraints related to the interdependence of activities were taken into account. Using many ACOs and cost management, a solution for task scheduling & load balancing was presented.Nonetheless, solitary occupations not suited to the network were already subject to non-restrictive management [9].

Systems or VMs which have been used at a capacity below their maximum capacity have been processed using load balancing. This result would be an error if the material requirements are used as much as possible [10, 11].

2. Related Works

Consequently, it was necessary to allocate the burden among the VMs to control and optimize both the makespan and the primary funding. A shorter makespan should not be associated with greater resource expenditures, and vice versa [12]. Physical virtual machines may assign and use resources [13]. It was likely that some virtual machines (VMs) would be overloaded while others would be undercooked when tasks were being carried out on VMs [14].

Because of the constant evolution of consumer demand, multi-leasing was necessary to keep different customers separate from cloud-based applications. Numerous heuristic and metaheuristic algorithms have been used to obtain efficient VM performance and distribute the workload among the available VMs [15]. This research project has already described and built a system for resource planning and load balancing. The proposed solution utilizes a combination of heuristic and metaheuristic methods to obtain the best results in terms of makespan and price [16].

Progress on mainframe processing to cloud technology was covered by A. Jain et al. Fundamental features, types, and designs of cloud applications were examined.Researchers also addressed various research and cloud technology issues.By modifying pheromone components and updating algorithms for pheromone components, this method maintains load equilibrium.While taking into account the embedded feature and load of each resource joint, others are trying to achieve load equality at the highest level.An innovative strategy to optimize ant colonies has been proposed by T. Liao et al. A probabilistic method for resolving computing issues was ACO. They employed ACO to schedule tasks for grid or cloud technology, but it didn't perform well because there are still some issues with pheromone generation & parameter estimation. Moreover, PACO encounters various issues, such as choosing parameters or getting pheromones.In this research, a self-adaptive improvement of the ant colony that improves the PACO was developed to enable the optimization of the ant colony to perform better.A method of task planning based on the gene ant colony algorithm was proposed by C. Y. Liu [4] for use in cloud technology. This method exploited the global discoverability of the GA to discover the best response and then turned it into the first pheromone of the ACO.The NP-hard nature of the task planning problem in the cloud computing system makes it suitable for intelligent optimization techniques to approximately identify the optimal response.Quality of Service (QoS) is a key measure for evaluating the effectiveness of task planning.Planning Cloud resources are crucial for delivering workflow in the software platform, according to X. F. Liu et al. [5], as regards the duration of implementation and operating expenses.They created an effective method based on the network of ant colonies to solve the challenge of optimizing implementation expenditures while meeting deadlines.An ant in the ACS symbolizes a strategy with T parameters for managing T activities on R resources (ACS). Compare the results obtained using the Dynamic Objective Genetic Algorithm (DOGA) and Particle Swarm Optimisation Methods (PSO). According to Jinhua Hu et al. [6], the planning of virtual machine (VM) assets in a cloud computing system primarily takes into account the program's current situation or infrequently takes into account framework alteration & historical information, which invariably results in an unbalanced load on the system. This work proposes a sequencing method for balancing VM resources based on the scalable method in light of the problem of balancing VM resources. The advances in cloud computing than BhathiyaWickremasinghe et al. [7] provide web app creators with a range of unique options. But there are not many resources available to help programmers evaluate the needs of massive Internet implementations from the difference in concentration of processing nodes or client workloads.

3. Proposed Methodology

Load balancing was done by the suggested HPOFT-MACO structure among the VMs to ensure that no VMs are overloaded or under-loaded. A similar workload has been assigned to enhance the use of virtual machines. Two distinct hybrid tasking strategies have been suggested and implemented for virtual machine workload management (VMs).The Forecast Completion Schedule Heuristic [17] for the current work was used to develop the initial workflow work plan, which was assigned to the VMs.

The production of ideal VM load balancing methods while satisfying many goals, such as reducing finishing time & boosting resource utilization, was possible by the combination of ACO with superior heuristics, such as POFT. The VM has been given responsibility for the tasks in the relevant processes to avoid delays in transmission between VMs [18]. A targeted operation is to be completed by the end of the waiting period, process activities have been prioritized based on their completion dates. In words, this algorithm uses constraints based on the quickest computation time needed for a project on a certain VM that the operation should finish completing before the timeframe arrives (see Figure 1).

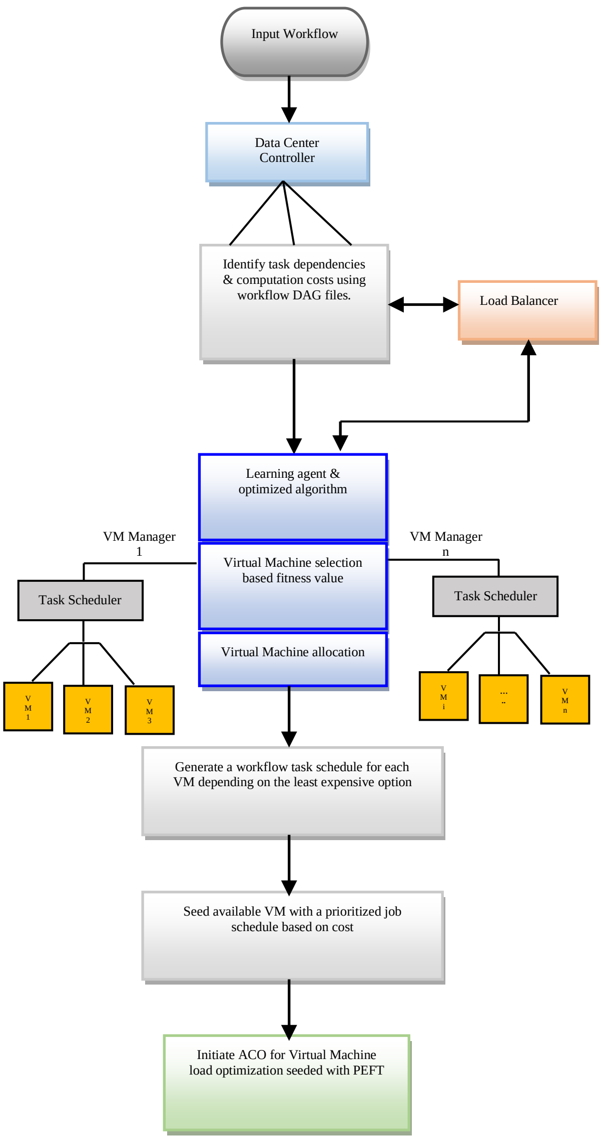

Fig. 1. POFT heuristics using ACO.

3.1 Propose HPOFT-MACO Approach

ACO was employed in this research to identify underutilized virtual machines (VMs) whose workload will be lower than the VM's lowest demand. This lower restriction of the VM limit was dynamically estimated to the very varied and dynamic requirement for running the cloud program.The first minimal workload on the virtual machine was determined by HPOFT. The burden that a physical computer running several virtual machines could manage was used to calculate those benchmark load values. Figure 2 illustrates how the criteria evolve as the number of VMs increases.

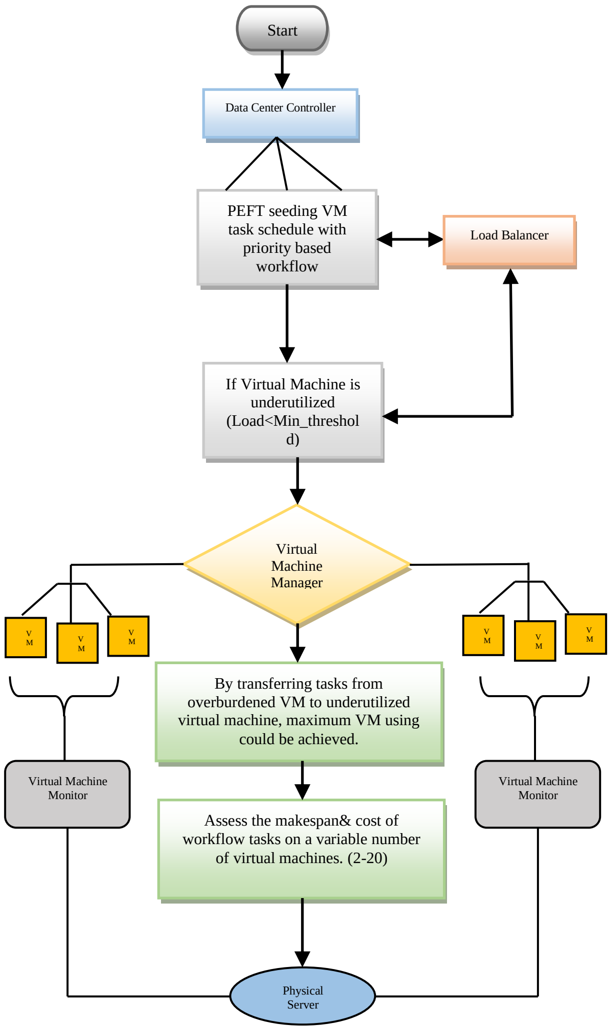

Fig. 2. HPOFT-MACO for VM Load Balancing.

Algorithm 1: Enhanced load balancing with HPOFT-MACO.

Step 1: Calculate the optimum cost (OC) for n jobs using the VM's ET- Estimation period & Ch- Child data.

Step 2: Use the formula $OC(t_n, v_k)$

$$OC(t_n, v_k) = \max_{t_1 \in ch_{t_n}} \left(OC(t_n, v_k) + ET(t_n, v_k) + Ch(t_n, v_1) \right)$$

to calculate $OC(t_n, v_k)$.

Step 3: Establish the min threshold of the formula VM = $OC(t_n, v_k)$ and use it to order operations.

$$Prior_{OC(t_n, v_k)} = \sum_{k=1}^{n} \frac{OC(t_n, v_k)}{nk}$$

end for

Step 4: For each 'P planning' assignment.

Step 5: Eliminate the top priority of ‘P Planning'.

Step 6: To use the formulae AFT$(t_n, v_k) = ET(t_n, v_k) + Ch(t_n, v_1)$& $O_{AFT}(t_n, v_k) = AFT(t_n, v_k) + OC(t_n, v_k)$ to calculate the average first time (AFT) - AFT(t_n, v_k)&optimized time OT(t_n, v_k)

7. Allocate $t_n$to VM to minimum O_{AFT}

Step 8: Add t_n the VM to the timetable, ordered by PriorOC

Step 9: Establish the minimal criterion and the schedule resource is important by the VM to seed the MACO.

Step 10: Population of ant, = Schedule by HPOFT

Step 11: A_n = Schedule $V_i$wherei =1

Step 11:i++

While ($i>i_{max}$)

end for

Step 12: end

A basic criterion was used to apportion the load, whose actual portion would be estimated by HPOFT heuristics. The processing capacity of the virtual machine had an impact on the importance of this criterion. When the demand for a VM is lower than a threshold set by MACO, the artificial ant finds an under-loaded VM in its neighbors and transmits its weight. This would be done after determining the altered dietary pheromone concentration. The amount of pheromone was initialized first, and then there are two replicates. The solution is built during the first iteration, or the number of pheromones has been updated during the second iteration.

A probability state transition strategy serves as the foundation for the ACO methods. Artificial ants could be considered stochastic greedy methods to construct solutions probabilistically by combining partial solutions with solutions obtained.Ants use pheromone streaks to record traits from the "excellent" alternatives created, which they use as a blueprint to build innovative solutions.The amount of pheromone has been updated using the heuristical information that would be obtained for each cycle.In addition, ants use problem-specific data to make decisions on how to come up with solutions.

4. Results and Discussions

In this task, the technology was increased to a total of 20 virtual machines (VMs).If the number of VMs were increased above this level, a larger makespan would be connected with an overload on the physical server.An adequate workload has been assigned to virtual machines as part of the current work to avoid the requirement to underload virtual machines.To simulate the offered HPOFT MACO, which divides a dedicated virtual server into a variable number of virtual machines, the Cloud Workflow Simulator (CWS) was used.The simulated physical system includes an 80 GB hard drive, 4 GB RAM, and an Intel (R) Core (TM) i5-4210U processor operating at 1.7 and 2.40 GHz.To duplicate a cloud environment with input process actions, CWS was perfect. The modeled results also imply that as the amount of VMs increases, more process activities were accomplished, resulting in a division of physical hosting resources among a large number of VMs and enhanced resource usage. This would occur if the quantity of VMs were increased beyond this cap; as a result, the one microprocessor would become overloaded and might stop functioning due to the heavy workload of workflow operations on the present VM. As the number of VM increases, process activities are expected to be completed.

The ideal workload for VMs and the number of processing techniques have been estimated, and VMs with burdens that are the ideal weight has been identified, guaranteeing that each VM has a constant workload. Two heuristics, HPOFT-MACO, were used for the initial planting of MACO.Makespan was measured in milliseconds, and to simulate different scenarios, the number of VMs running on hosts was completely changed from 2 to 20.For each job performance, the average makespan was determined using the two-hybrid techniques. After using a hybrid HPOFT-MACO technique to perform the Cybershake procedure on 18 VMs, Table 1 shows a screenshot of the CWS's source information. It was 100 measures in total.The findings of the average length and expenditure of the 100 replicates were considered for some additional research.We looked at the extent to which VMs varied.Using 18 VMs, the results in Table 1 were reached.Suggested hybrid solutions are incapable of completing a process that does not respect time restrictions.The parameters 0 makespan& value in the illustration showed that (see Table 1).

Table 1. Modelled results.

S.No	Application	Distributed	Algorithm	Cost	Lastdagfinish
1	GENOME	HPOFT	MACO	91	27627.56
2	GENOME	HPOFT	MACO	85	29825
3	GENOME	HPOFT	MACO	86	56752
4	GENOME	HPOFT	MACO	95	54192.35
5	GENOME	HPOFT	MACO	76	54192.35
6	GENOME	HPOFT	MACO	83	54192.35
7	GENOME	HPOFT	MACO	25	54192.35

8	GENOME	HPOFT	MACO	19	54192.35
9	GENOME	HPOFT	MACO	21	54192.35
10	GENOME	HPOFT	MACO	22	54192.35
11	GENOME	HPOFT	MACO	26	54192.35
12	GENOME	HPOFT	MACO	99	23584.36
13	GENOME	HPOFT	MACO	110	42670.648
14	GENOME	HPOFT	MACO	789	86483.98
15	GENOME	HPOFT	MACO	761	56454.62
16	GENOME	HPOFT	MACO	167	48917
17	GENOME	HPOFT	MACO	358	67848
18	GENOME	HPOFT	MACO	890	77980.84
19	GENOME	HPOFT	MACO	720	98479
20	GENOME	HPOFT	MACO	462	86871.39
				⋮	
41	GENOME	HPOFT	MACO	647	26578.45
42	GENOME	HPOFT	MACO	367	98476.65
43	GENOME	HPOFT	MACO	829	68415.33
44	GENOME	HPOFT	MACO	798	23561.72
45	GENOME	HPOFT	MACO	560	65489.02
46	GENOME	HPOFT	MACO	949	89889.05
47	GENOME	HPOFT	MACO	583	35641.65
48	GENOME	HPOFT	MACO	638	46867.64
49	GENOME	HPOFT	MACO	679	8862.56
50	GENOME	HPOFT	MACO	790	36574.69
			Average	323.67	67547.64

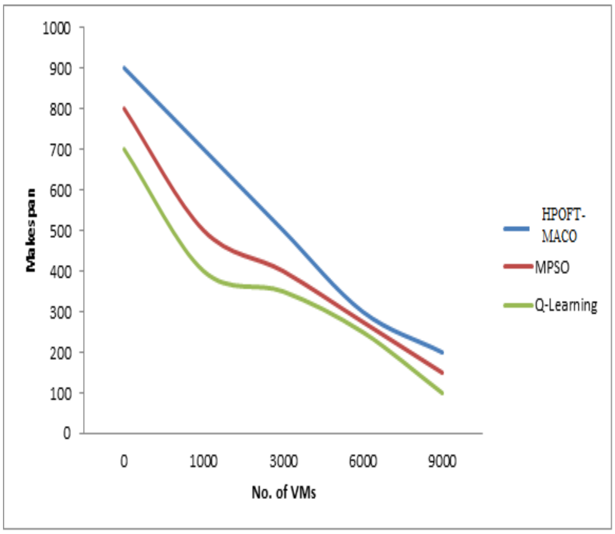

Fig. 3. No. of VM vs Makespan.

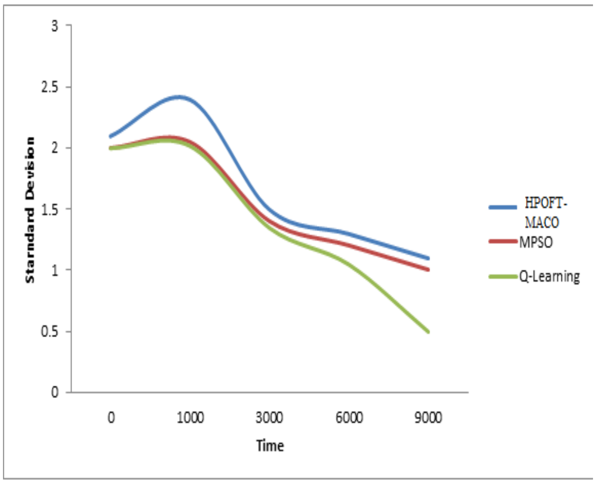

Fig. 4. Time vs SD.

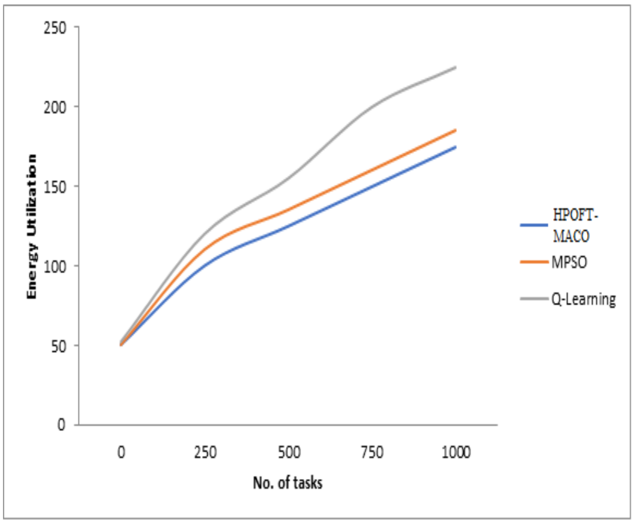

Fig. 5. No. of tasks vs Energy.

Other situations have been the subject of investigations into various VM accounts.The fact that deployment outcomes for 100 iterations of the three workflows using the two methodologies provided varied widely.Consequently, the research illustrated in Table 2 used the average for each deployment cycle in several operations.

Table 2. Comparison of results of VM.

Number of VM	HPA	HHA
2	0	2696.34
4	323.45	19962.612
6	4368.03	6478.23
8	694.232	82532
10	915.64	6693.93
12	1254.27	5842.61
14	1591.923	6085.12
16	1622.55	5908.85
18	1732.86	5046.73
20	2141.642	4135.11
Average Makespan	1222.53	9321.241

Fig. 6. HPOFT-MACO Scheme.

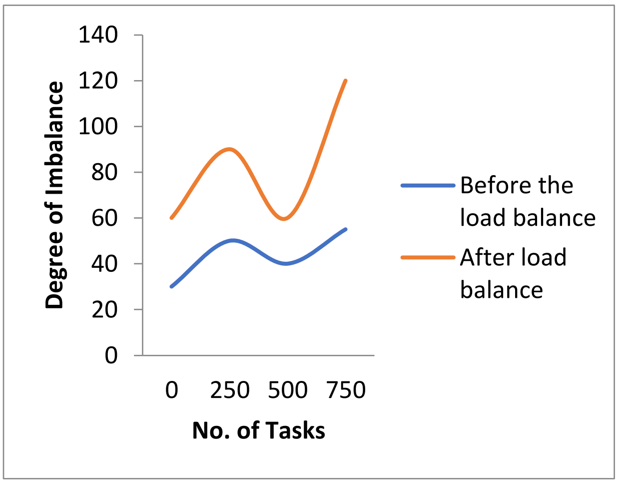

Fig. 7. Load balancing before and after.

Ligo analyses the inputs from Figure 7 as the task execution through the proposed HPOFT-MACO framework, a pattern similar to the Cybershake procedure has been observed in the makespan comparison between HPOFT-MACO approaches. Both VMs are used as assets to implement the Ligo process, the makespan of the proposed method using HHA would be approximately 200 times longer than HPA.The planning discrepancy decreases and becomes almost equal when 10, 12, 14, 16, 18, or eventually 20 VMs are used for Ligo's calculation. According to the findings acquired during the execution of the three primary workflows on the suggested HPOFT-MACO, the HPOFT-MACO method produced better average outcomes than the hybrid HEFT-ACO method.

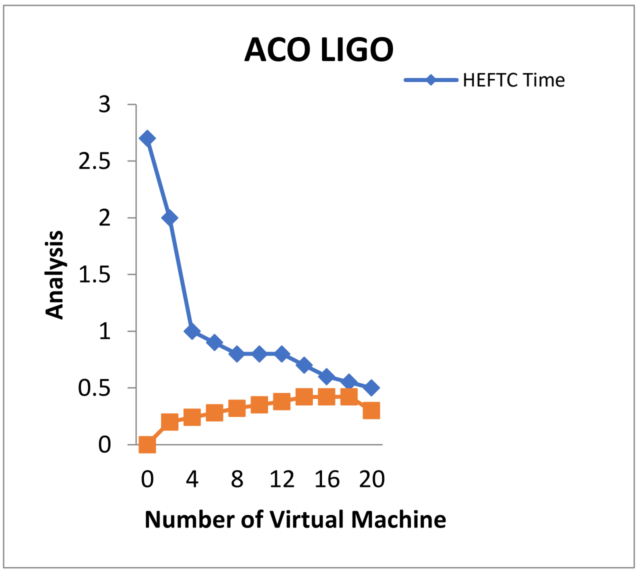

Fig. 8. Loch Ligon HPA and HHA Techniques**.**

Cost analysis

The proposed HPOFT-MACO hybrid efficacy evaluation process has been evaluated here based on a price indicator. This was one of the most critical qualities of service factors, as cloud-based services were provided on a pay-per-use basis.Both the system administrator and the consumer should minimize economic effects. The findings of the HHA installation method for Cybershake reveal a considerable improvement among 2 and 4 VMs and a further reduction from 6 VMs. Figure 9 compares the costs of the two approaches employed in the Cybershake HPOFT-MACO process.The results of HPOFT-MACO show a continuous trend of cheap prices with a small number of virtual machines increasing rapidly as the quantity of virtual machines increases. The price differences between the two-hybrid approaches would be perceptible when the amount of VM was small. The expense difference between the two methods was Rs 81.35 when just 2 VMs were used, but when the number of VMs rises, The price difference decreases till it reaches Rs 8.96 once 20 VMs are utilised in the proposed architecture. (see Table 3).

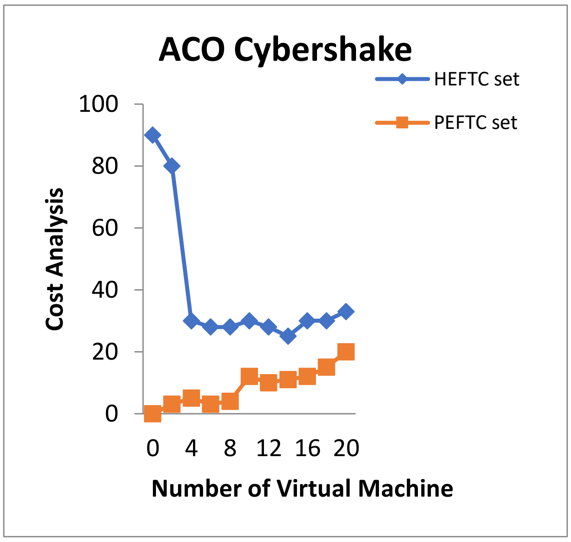

Fig. 9. Compares the costs of the HPA and HHA techniques.

Table 3. Compares the prices of VM HPA and HHA methods.

Number of Virtual Machine	**Cybershake**	
	HPA	**HHA**
2	0	81.37
4	1.22	63.38
6	4.73	30.16
8	3.83	28.33
10	4.75	27.64
12	9.91	30.69
14	9.09	28.63
16	11.54	25.12
18	15.45	28.9
20	20.22	29.53
Average Makespan	8.047	38.312

Initially, it was determined that the price gap between HPA and HHA techniques was rather substantial for 2 and 4 VMs, however this disparity steadily reduced as the number of VMs increased. According to the findings of HHA with Cybershake, the performance of two to four VMs increased dramatically before being optimised for six VMs.The results of the HPA technique show a constant trend of low costs with a small number of VMs, then prices increase steadily as the quantity of VMs increases.Both strategies produce similar results in that fewer virtual machines cost less, and the number of VM increases, as well as the average cost.The HPOFT-MACO suggested using the HPA technique has the lowest average cost of the three processes when the Cybershake procedure was used.

5. Conclusion

This study presents several gaps in the historical literature for task scheduling using meta-heuristic optimization strategies.A framework for VM job planning called "Time-Constrained Hybrid Approach, Variable VM Generation & Load Balancing" has already been designed and deployed to address the highlighted constraints.Two-hybrid approaches, POFT-ACO and HEFT-ACO, were employed to build an extensive architectural style.This study presents several gaps in the historical literature for task sequencing using metaheuristic optimization techniques.A VM job planning model called "time-constrained, Dynamic VM Supply & Load Balancing Hybrid Centric Approach" has already been developed and tested to address the highlighted restrictions.An entire application is implemented using hybrid HPOFT-MACO methods. In addition, the results of the Cybershake&Ligo workflows were more focused on improving and reducing effort and expense, but the results of the genome process took much more effort and money. Results from this study indicate that the suggested HDD-PLBFW infrastructure produces the best Cybershake&Ligo results for the HPOFT-MACO approach.

Author contributions

Prabhakara B. K: Conceptualization, Methodology, Software, Field study, Writing-Original draft preparation, Validation,

Dr. Chandrakant Naikodi: Visualization, Investigation, Writing-Reviewing and Editing.

Dr.Suresh L: Visualization, Investigation, Field study

Conflicts of interest

The authors declare no conflicts of interest.

References

AnniePoornima Princess, G., &Radhamani, A. S. A hybrid meta-heuristic for optimal load balancing in cloud computing. *Journal of Grid Computing*, *19*(2), 1-22. (2021).

Sridevi, G., &Chakkravarthy, MA meta-heuristic multiple ensemble load balancing framework for the real-time multi-task cloud scheduling process. *International Journal of System Assurance Engineering and Management*, *12*(6), 1459-1476. . (2021).

KakkottakathValappilThekkepuryil, J., Suseelan, D. P., &Keerikkattil, P. M An effective meta-heuristic-based multi-objective hybrid optimization method for workflow scheduling in a cloud computing environment. *Cluster Computing*, *24*(3), 2367-2384. . (2021).

Malathi, K., &Priyadarsini, K. Hybrid lion–GA optimization algorithm-based task scheduling approach in cloud computing. *Applied Nanoscience*, 1-10. (2022).

Pradhan, A., Bisoy, S. K., & Das, A. A survey on ACO-based meta-heuristic scheduling mechanism in a cloud computing environment. *Journal of King Saud University-Computer and Information Sciences*. (2021).

Chaudhury, K. S. A Particle Swarm and Ant Colony Optimization based Load Balancing and Virtual Machine Scheduling Algorithm for Cloud Computing Environment. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(11), 3885-3898. (2021).

Ziyath, S., &Senthilkumar, S MHO: metaheuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services. *Journal of Ambient Intelligence and Humanized Computing*, *12*(6), 6629-6638. . (2021).

Krishnmoorthy, P. Performance Analysis of Hybrid BAT Algorithm and Cuckoo Search Algorithm [HB-CSA] for Task Scheduling in Mobile Cloud Computing. *Available at SSRN 3997784*. (2021).

Kodli, S., &Terdal, S. Hybrid max-min genetic algorithm for load balancing and task scheduling in a cloud environment. *Int J IntellEng Syst.*, *14*(1), 63-71. (2021).

Chaudhury, K. S. A Particle Swarm and Ant Colony Optimization based Load Balancing and Virtual Machine Scheduling Algorithm for Cloud Computing Environment. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(11), 3885-3898. (2021).

Saoud, A., &Recioui, A. A hybrid algorithm for a cloud-fog system-based load balancing in smart grids. *Bulletin of Electrical Engineering and Informatics*, *11*(1). (2022).

Suresh, S., &Sakthivel, SA novel performance constrained power management framework for cloud computing using an adaptive node scaling approach. *Computers & Electrical Engineering*, *60*, 30-44. . (2017).

Ullah, A., &Nawi, N. M. An improvement in tasks allocation system for virtual machines in cloud computing using HBAC algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1-14. (2021).

Kannammal, A., & Suresh, S. A hybrid approach-based energy-aware cluster head selection for IoT application. In *Inventive Communication and Computational Technologies* (pp. 563-568). Springer, Singapore. (2020).

Talouki, R. N., Shirvani, M. H., &Motameni, H A hybrid meta-heuristic scheduler algorithm for optimization of workflow scheduling in a cloud heterogeneous computing environment. *Journal of Engineering, Design, and Technology*. . (2021).

Albert, P., &Nanjappan, M. WHOA: Hybrid Based Task Scheduling in Cloud Computing Environment. *Wireless Personal Communications*, *121*(3), 2327-2345. (2021).

Kumar, R., &Bhagwan, J. A Comparative Study of Meta-Heuristic-Based Task Scheduling in Cloud Computing. In *Artificial Intelligence and Sustainable Computing* (pp. 129-141). Springer, Singapore. (2022).

Suresh, S., &Sakthivel, S. System modeling and evaluation of factors influencing power and performance management of cloud load balancing algorithms. *Journal of Web Engineering*, 484-500. (2016).