International Journal of Computer and Engineering Optimization (IJCEO)
Volume 1, Issue 1, September – October (2023)

RESEARCH ARTICLE

C-AVPSO: DYNAMIC LOAD BALANCING USING AFRICAN VULTURE PARTICLE SWARM OPTIMIZATION

Dharavath Champla[1,*], V. Ramkumar [2] and P. Ajay[3]

[1]Department of Computer Science and Engineering, SRM Easwari Engineering college, Chennai.
[2] Department of Electronics and Communication Engineering, K. Ramakrishnan college of Technology, Samayapuram, Tamil Nadu 621112, India.
[3]Department of Electronics and Communication Engineering, Rathinam Technical Campus, Eachanari, Coimbatore, Tamil Nadu, 641021 India

Corresponding e-mail: champla.805@gmail.com

Abstract – A new technology called cloud computing enables users to access services from anywhere, at any time, under different conditions, and is controlled through an outside cloud service provider. Cloud task scheduling is a complicated optimisation problem. However, both under- and over-loading conditions cause a range of system problems as far as power consumption, machine failures, and so forth are concerned. Consequently, virtual machine (VM) work-load balancing is regarded as a key element of cloud task scheduling. In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed. Using C-AVPSO, the developed optimization algorithm solves the dynamic load balancing problem effectively. This approach used the AVO process to get the exploration space, while the increased response was identified by the PSO procedure. This algorithm successfully addresses the task's resource utilization, response time, and budgetary restrictions. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, in the cloud context, load balancing performance measures and convergence rate are enhanced. To enhance the operation's efficiency, the proposed method balances VM loads efficiently. The proposed framework is compared to existing approaches like QMPSO, FIMPSO and ACSO Based on energy utilization, degree of imbalance and task migration, response time and resource utilization. The proposed C-AVPSO technique reduces resource utilization of 19.1%, 31%, and 54% than, QMPSO, FIMPSO and ACSO existing techniques.

Keywords – *DYNAMIC Load Balancing, swarm optimization, PSO algorithm, Virtual machine, Cloud computing.*

1. INTRODUCTION

Cloud computing is the most advanced and rapidly evolving technology in computer science today [27]. The cloud is a network of IT resources, and computing is the act of executing work in remote connection to those resources and charging a pay-as-you-go system. A technology based on the internet that offers a variety of cloud-based services that are efficient, dependable, and inexpensive, [29] and these services may be accessed from any device, location, or time. It offers on-demand self-service, which means that when we need a resource, we can use (configure) it without requiring the assistance of a third party. Today, there are other cloud service providers to choose from, such as Google Cloud and Amazon Web Services. A computer model called "cloud computing" gathers resources over the Internet. [28] For example, servers, storage, applications, and services.

By ensuring sufficient cloud resource management, the affordable and scalable benefits of cloud computing may be realized. One of the major elements of the cloud structure is that these cloud resources are virtual. Customers can rent services of Cloud Service Provider (CSP) offerings [30]. With resources for the virtual cloud at hand, the CSP's role in providing services to the user is highly complex. As a result, load balancing has received more attention from researchers. This load balancing improves overall system performance. Cloud Service Providers (CSPs) are left with unbalanced computers that have a wide Resources and tasks gradients of user's consumption as a result. [11]

No machine is overworked or underpowered thanks to Redistributing workloads in distributed systems like the cloud [12,13]. The technique of load balancing has assisted networks and resources in delivering the highest throughput with the quickest reaction times. [14In load balancing, a number of factors are accelerated to improve the performance of the cloud, such as reaction time, execution time, and system stability. [15,16]. Several academics have discussed load balancing strategies, such as (i)static load balancing and (ii) dynamic load balancing, in both heterogeneous and homogeneous situations. [17]

When the single VM is overloaded with tasks and if there are a lot of empty virtual machines in the cloud network, it would be best to move the workloads from the overworked to the underworked ones. [26]. Calculating every conceivable task-resource mapping in a cloud context is challenging, and finding the best mapping is not an easy process. [24] Thus, we require an effective task distribution method that can schedule tasks in a way that prevents a large number of virtual computers from being overburdened or underloaded. The cloud task scheduler [25] then begins to perform load balancing operations as soon as it has allocated the task to a virtual machine, so that tasks can be transferred between overloaded and underloaded virtual machines after the task has been allocated to a virtual machine while maintaining the balance of all virtual machines.

This is an overview of this paper's main contributions;

- In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed.
- The C-AVPSO optimization algorithm efficiently balances load in cloud networks. In this technique, the AVO process was used to acquire the exploration space, and the PSO procedure was used to identify the improved response.
- In the developed algorithm, the constraints related to resource utilization, response time, and cost are successfully resolved.
- In a cloud environment, C-AVPSO improves the convergence rate and performance metrics by combining AVO and PSO algorithms. This method maximizes operation efficiency by efficiently balancing VM loads.
- A comparison of the proposed framework and existing approaches like QMPSO, FIMPSO and ACSO is conducted based on energy consumption, degree of imbalance, migration of tasks, response time, and resource usage.

Therefore, the remainder of this article will be structured as follows: The first part of the paper provides a review of the literature. Following that, the proposed research is evaluated in Section 3, results are discussed in Section 4; finally, a conclusion is presented in Section 5.

2. LITERATURE SURVEY

For load balancing in CC, a variety of heuristics and meta-heuristic methods have been used. This section summarizes the pertinent work in these areas with a particular emphasis on African vulture particle swarm optimization (AVPSO) for dynamic load balancing. In this part, we've talked about a few of those technique.

In 2020 Mishra, S.K. et al. [1] proposed a category of algorithms for cloud load balancing. Additionally, several approaches to load balancing in cloud computing platforms are explained. Finding overloaded and under loaded nodes and balancing the load are the steps in the load balancing process. between them. The simulation is performed in clouds simulator to examine the Heuristic-based performance methods, and the results are detailed.

In 2019 Afzal, S. and Kavitha, G., [2] proposed a comprehensive encyclopaedic analysis about the load balancing techniques. The pros and disadvantages of the current methods are described, and algorithms are addressed. As a result, 80% of works do not analyze how the load balancing algorithm performs while evaluating performance.

In 2018 Volkova, V.N. et al. [3] proposed the cloud analyst analytical tool is used to assess various algorithms. A comparison of algorithm load balancing algorithms is also performed. Load balancing helps the centralized server run better. The load balancing algorithm investigated. Results were compared using Data on total response time, center time, and data center load and processing on an hourly basis cost.

In 2022 Jena, U.K., et al. [4] proposed QMPSO is a revolutionary methodology for dynamic load balancing across virtual machines that uses a mixture of an enhanced Q-learning algorithm and amended Particle Swarm Optimization (MPSO). Hybridization's goal is to improve machine performance through load balancing among the virtual machines. The algorithm's resilience was demonstrated by comparing the QMPSO simulation results to the current load balancing and scheduling technique.

In 2019 Polepally, V. and Shahu Chatrapati, K. [5] proposed a load-balancing technique based on constraint measure. First, the load and capacity of each virtual machine are calculated. The load balancing approach computes and analyses the decision factor for each virtual machine. The suggested load balancing method's performance is compared to those of current load balancing techniques like HDLB, DLB, and HBB-LB for capacity and load estimation parameters.

In 2022 Latchoumi, T.P. and Parthiban, L. [6] proposed to obtain the best resource scheduling in a CC scenario, an innovative Quasi Oppositional Dragonfly Algorithm for Load Balancing (QODA-LB) was developed. The main goal of this strategy is to decrease task execution costs and times while keeping the load distributed evenly across all VMs in the CC system. The simulation results showed superior performance to the leading methods and optimal load balancing efficiency.

In 2020 Devaraj, A.F.S., et al. [7] proposed, Firefly and the Improved Multi-Objective Particle Swarm Optimization (FIMPSO,) as a new load balancing algorithm. According to the simulation results, the FIMPSO algorithm produced the most efficient end with shortest common response time of 13.58ms, surpassing all other comparable techniques with the greatest CPU utilization of 98%, the highest memory utilization of 93%, the highest dependability of 67%, the greatest throughput of 72%, and the maximum make span of 148.

In 2020 Semmoud, A., et al. [8] proposed a fresh method for various cloud computing configurations for load balancing. The suggested method seeks to reduce Makespan and VM idle time while raising system stability.When the VM load surpasses the Starvation Threshold, an adaptive limit, the suggested method restricts task transfer. We compared the STLB algorithm to a load balancing algorithm that was inspired by honey bee behaviour, and we found that the suggested method outperformed about the amount of migrations and the typical idle time, the HBB-LB algorithm.

In 2021 Balaji, K., et al. [9] proposed, a load balancing system that addresses optimization suggested by using the adaptive cat swarm optimization (ACSO) method. The effectiveness of the suggested method is assessed with a variety of value indicators, and its performance is contrasted with that of competing techniques. Our suggested solution takes the least time and energy in compared to the current algorithm.

In 2017, Kumar, M. and Sharma, S.C. [10] proposed a load-dynamic balancing method that speeds up cloud resource use while decreasing make-span time. a conventional approach using Cloud load balancing through task migration. In comparison to FCFS and SJF approaches, the results of the trial show that the recommended method lowers the manufacture span time and boosts the average resource utilisation ratio.

It can be seen from the reviews above that these methods have some shortcomings. This research proposes a AVPSO technique for dynamic load balancing to address these disadvantages.

3. PROPOSED METHODOLOGY

This article presents a new algorithm for African vulture particle swarm optimization that optimizes dynamic load balancing by considering cost, resource use, and response time. By employing this technique, you may balance job preferences, boost the throughput of the virtual machines, and disperse the load among them by adjusting the waiting times for complicated tasks.

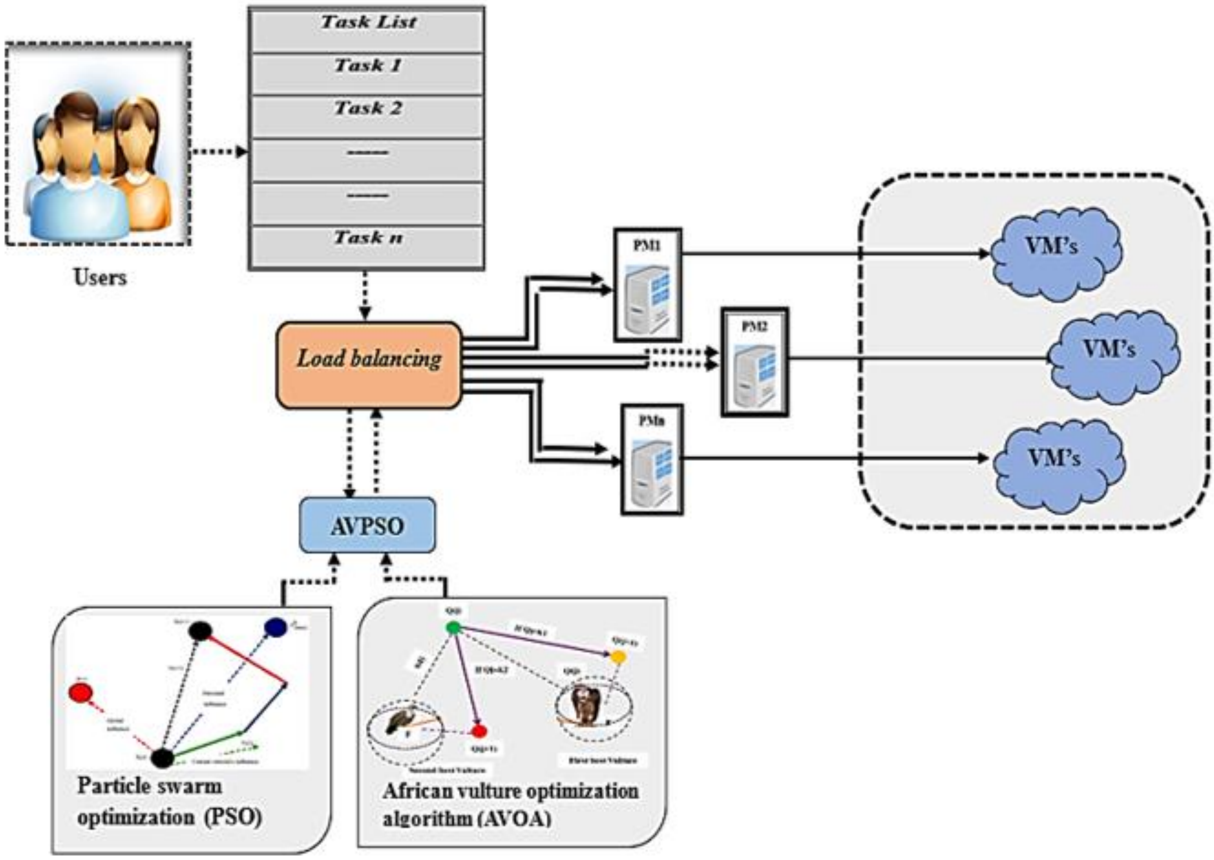

Figure 1. Block diagram of proposed methodology

Figure.1 illustrates the proposed methodology. The primary process in the cloud is workload distribution to virtual machines (VMs). Using node performance data, decisions about load balancing happen in the dynamic load balancing mode. Due to low number of VMs, resource delivery to the task is crucial in cloud systems.

As a result of the VM being overburdened with jobs, the reaction time of the system is lengthened. Thus, a dynamic load-balancing procedure based on AVPSO is suggested to distribute the tasks across the virtual machines (VMs). This strategy involves moving Underloaded VMs take on the VMs' burdened tasks. As a result, both the latent times and the performance are accelerated. The recommended work-based load balancing method distributes the load in the cloud efficiently. The system increases resource utilization while reducing costs and response times. The capacity of each VM is determined after job scheduling. Divide the overflow, underload, and balanced containers according to the VM's remaining capacity. To complete the load balancing procedure, the ideal underload container is found in the proposed task. The tasks are then moved using the migration approach of the best under loaded virtual machines to the worst overloaded ones. The AVO algorithm is used into PSO to finish the load balancing process and obtain the search space.

3.1. Load balancing in cloud

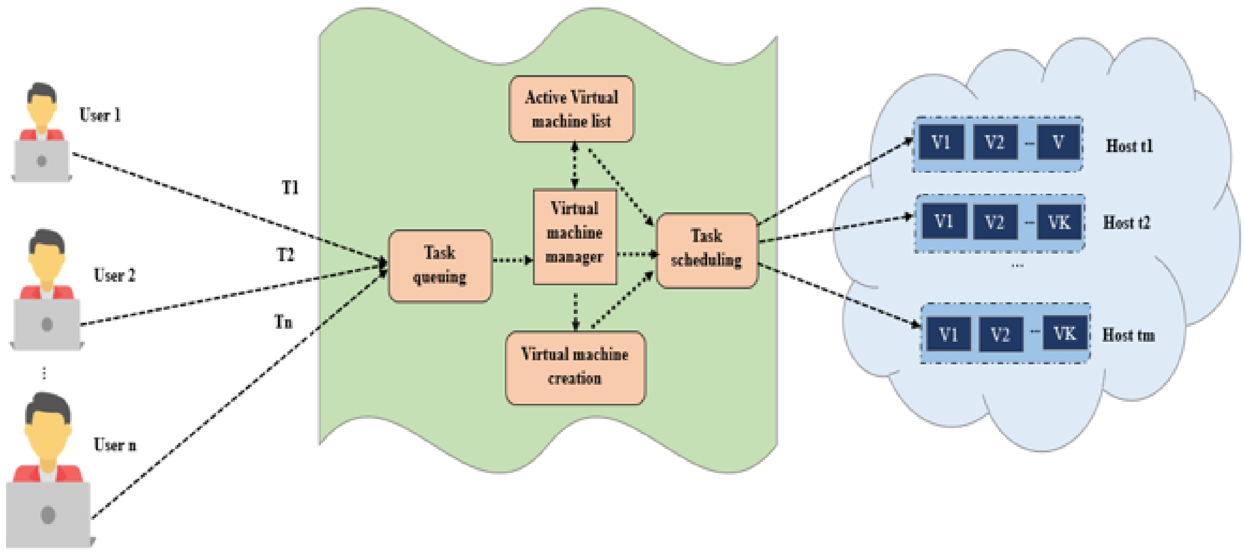

Figure 2. Scheduling cloud tasks for load balancing

Figure 2 depicts the scheduling task in cloud. Owing to cloud computing, that offers cloud services to cloud consumers, every operation is completed in a cloud environment. Due to the enormous volume of diverse input tasks so as to balance the demands of diverse resources, load balancing is necessary. The cloud system's task queue receives the tasks with n inputs $T_1, T_2, \dots T_n$ .The Virtual Machine head then the get input tasks from task queue and have a comprehensive knowledge of the active VM, existing resources across servers, and the length of the local task queue across all hosts. The system's resource availability was confirmed by the VM manager. The VM management submitted the tasks to the task scheduler if the group of tasks could be completed using the active VMs that are now available. If resource availability does not meet requirements, the VM management generate the necessary VMs in the server. Task allotment in cloud computing is therefore quite difficult. The service's QoS degrades when only a small number of VMs are overloaded, only a small number are free, or when there are fewer tasks to complete. Users may switch to another Cloud provider if they are unhappy with their current service as a result. Each cloud server is limited in the amount of virtual machines it can support.

3.2. Particle Swarm Optimization (PSO)

Particle swarm optimisation (PSO), is one of the bio-inspired algorithms, is basic in its quest to find the supreme answer in the area of problems. It is distinct from conventional optimisation methods in that it only uses the objective function itself and does not depend on the gradient or any differential forms of the objective function. The search is impacted by two distinct learning processes carried out by the particles in PSO. Each particle learns from its own movement-related experiences as well as those of other particles. Learning from one's own experiences is referred to as cognitive learning, whereas social learning involves learning from others. Using social learning, each swarm particle visits the best solution, which is then recorded in each particle's memory as *gbest*. The particle stores the best solution it has independently discovered so far, known as *pbest*, in its memory through cognitive learning. In terms of PSO, time is the iteration. The rate at which the position is changing in relation to the iteration can be regarded of as the velocity in PSO. The iteration counter increases by a factor of unity, which leads to equalize velocity V and position X, the dimensions must be the same.

The most efficient response for a D-dimensional search space, with the i^{th} particle of the swarm at the step time t denoted by a D- dimensional vector., $x_i^t = (x_{i1}^t, x_{i2}^t, \dots . x_{iD}^t)^T$. Likewise, the velocity at step time t can be represented by another D−dimensional vector $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)^T$.

The earlier position of the i^{th} particle at the step time t is denoted as $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)^T$. $'g'$indicates which particle is the most efficient in swarm. Using velocity update equation, the i^{th} particle's velocity is upgraded in equation (1)

Velocity update equation:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (1)$$

As shown in (2), the position is upgraded based on the position update equation;

Position update equation:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

Where, $d = 1,2, \dots D$ denotes the dimensions where the values i=1,2,...,s stand for the particle index. Whereas c1 and c2, or the cognitive and social scaling factors, are constants, S is the size of the swarm. It seems from equations (1) and (2) that each particle's dimensions are changed individually. Through the locations of the top places, *gbest* and *pbest*, so far discovered. Equation (1) and (2) outline the PSO algorithm's fundamental configuration. Algorithm provides a PSO method algorithmic approach.

Algorithm 1

Create a D-dimensional swarm which has been initialized with the velocity vectors associated with it;

for t=1 to the *maximum bound pn the number of iterations* **do**

for i=1 to S **do**

for d=1 to D **do**

Implement equation 1 to update velocity;
Implement equation 2 to position velocity;

end

Evaluate updated location fitness;

Prior data on gbest and pbest be updated on necessity;

end

Stop when the problems solved by gbest;

end

3.3 African Vulture Optimization algorithm (AVOA)

The algorithm, known as the AVOA illustrates how African vultures navigate and forage for food. African vultures are among the rare species of vultures capable of reaching the maximum altitude among the various vulture species. African vultures are continually moving from one place to an additional in pursuit of better food sources due to their circular motions in the sky. They are at odds with one another in order to obtain the food source. The initial vultures used by the AVO algorithm are some random individuals, and after determining their objective value, their ability is calculated. Each time, one of the top two vultures is either moved or eliminated by a new population type. The following is a list of the prerequisites and points for the regular AVOA.

$$R_i = \begin{cases} Best\ volture\ 1, & if\ pi = L_1 \\ Best\ volture\ 1, & if\ pi = L_2 \end{cases} \quad (3)$$

$$L_1 + L_2 = 1 \quad (4)$$

where,

L1 and L2 define two parameters that are attained before optimisation in the range [0, 1]. to decide which group member is the finest,

$$Pi = \frac{F_i}{\sum_{j=1}^{m} F_i} \quad (5)$$

In equation (5) 'F' determines the vultures' level of contentment,

The ratio of vulture starvation has then been determined. As a person runs out of energy, they will engage in combat with nearby, more powerful vultures to obtain food. You can model this as follows:

$$t = k \times \left(sin^w\left(\frac{\pi}{2} \times \frac{iter_i}{max_{iter}}\right) + cos\left(\frac{\pi}{2} \times \frac{iter_i}{max_{iter}}\right) - 1\right) \quad (6)$$

$$F = (2 \times \delta_1 + 1) \times y \times \left(1 - \frac{iter_i}{max_{iter}}\right) + 1 \quad (7)$$

Equation (6) uses w to denote a constant to represent an optimisation procedure, and $iter_i$ to denote the current iteration. y represents a randomly inserted value among 0 and 1. k specifies the random value in range of [2, 2], and δ_1 denotes a random integer between 0 and 1. max_{iter}defines the total number of iterations. The vulture becomes hungry if y decreases to 0, else, it increases to 1.

After that, a random mechanism with two policies was taken into consideration to execute algorithm exploration. The following are examples of how people in an environment hunt for food sources:

If P_1 is less than $rand_{p1}$,

$$P(i+1) = R_i - F + \delta_2 \times ((ub - lb) \times \delta_3 + lb) \quad (8)$$

If P_1 is above or equal to $rand_{p1}$,

$$P(i+1) = R_i - D(i) \times F \quad (9)$$

Where,

$$D(i) = |X \times R(i) - P(i)| \quad (10)$$

R denotes a supreme vulture, X indicates how the vulture decides whether or not to keep food acquired from another vulture, which is obtained by $X = 2 \times \delta_i$ where $i = 1,2,3$ two numbers that are created randomly in the value of [0, 1], and $ub\ and\ lb$ denote the boundaries for variables at both lower and higher levels.

Additionally, $|H|$ should be less than 1 in order to abuse the algorithm. This consists of two parts with two siege-fight and rotating flight policies, defined by P_2 and P_3 as two parameters ranging from 0 to 1. Based on the strategy described above, the weakest vulture tries to steal the healthiest food in specified manner that follows;

$$P(i+1) = D(i) \times (F + \delta_4) - d(t) \quad (11)$$

$$d(t) = R_i - P(i) \quad (12)$$

Where, δ_4 is a probability number between 0 and 1.

Moreover, the following is the mathematical description of the vulture's spiral motion:

$$S_1 = R(i) \times \left(\frac{\delta_5 \times P(i)}{2\pi}\right) \times \cos(P(i)) \quad (13)$$

$$S_2 = R(i) \times \left(\frac{\delta_6 \times P(i)}{2\pi}\right) \times \sin(P(i)) \quad (14)$$

$$P(i+1) = R_i - (S_1 + S_2) \quad (15)$$

where δ_5 and δ_6 represent two random numbers between "0" and "1." Most vultures will struggle for food in the beginning if δ_{p_3} ,is a random number between 0 and 1, it's bigger than (or equal to) P_3. The harsh siege-fight policy has been used if δ_{p_3} is less than P_3. When vultures are famished, it can create a huge competition among them to locate food. The following equation accomplishes this:

$$A_1 = BestVilture_1(i) - \frac{BestVilture_1(i)\,(i) \times P(i)}{BestVilture_1(i)\,(i) - P(i)^2} \times F \quad (16)$$

$$A_2 = BestVilture_2(i) - \frac{BestVilture_2(i)\,(i) \times P(i)}{BestVilture_2(i)\,(i) - P(i)^2} \times F \quad (17)$$

where, $BestVilture_1(i)$ *and* $BestVilture_2(i)$ represent the best vultures from both sets, while P(i) represents the vector's position in the moment.

$$P(i+1) = \frac{A_1 + A_2}{2} \quad (18)$$

The once-healthy vultures lose their strength and capacity to speak in front of crowds. They then fly to a different location to acquire food once more,

$$P(i+1) = R(i) - |d(t)| \times F \times LF(d) \quad (19)$$

where, LF denotes Levy flight (LF) and calculated analytically as follows:

$$LF(x) = \frac{u \times \sigma}{100 \times |v|^2} \quad (20)$$

$$\sigma = \left(\frac{\Gamma(1+P) \times \sin\frac{\pi\rho}{2}}{\Gamma(1+\rho_2 \times \rho \times 2\left(\frac{\rho-1}{2}\right)} \right) \quad (21)$$

Where, ρ denotes the fixed value, while u and v are the arbitrary numbers between 0 and 1.

4. RESULTS AND DISCUSSIONS

This algorithm is implemented in Cloud Sim as a load balancing algorithm based on C-AVPSO. Our proposed method is similar to the traditional methods QMPSO, FIMPSO, ACSO in terms of the energy utilization, degree of im-balance, number of tasks migration, response time and resource utilization.

4.1 Evaluation metrics

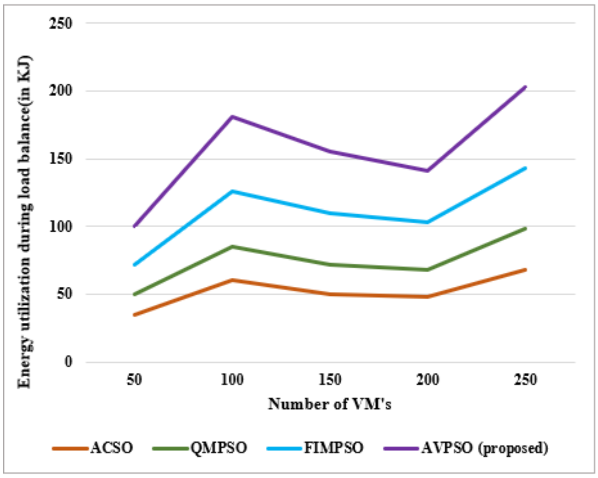

Figure 3. Energy utilization vs number of VM's

4.1.1. Energy utilization

When compared to other algorithms like FIMPSO, ACSO, and QMPSO during load balance, the proposed C-AVOPSO technique used the most energy. Also, the energy utilisation analysis revealed that, when compared to other algorithms, the proposed C-AVPSO approach required the least amount of energy (by altering the number of VMs from 0 to 250).

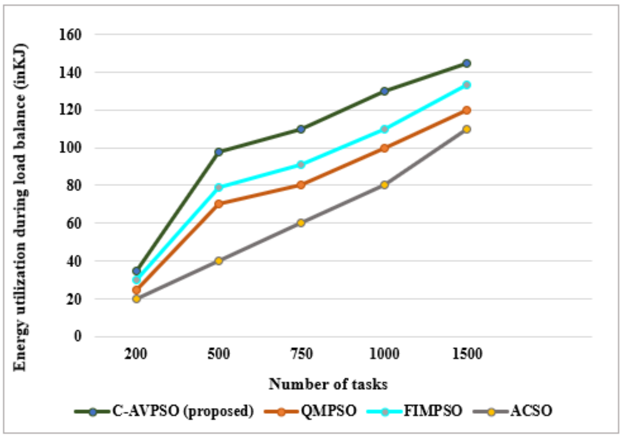

Figure 4. Energy utilization vs number of tasks

By comparing the suggested AVPSO to the various current algorithms, it was discovered that the proposed AVPSO used the most energy within number of Tasks from 100 to 1500.

4.1.2. Migration

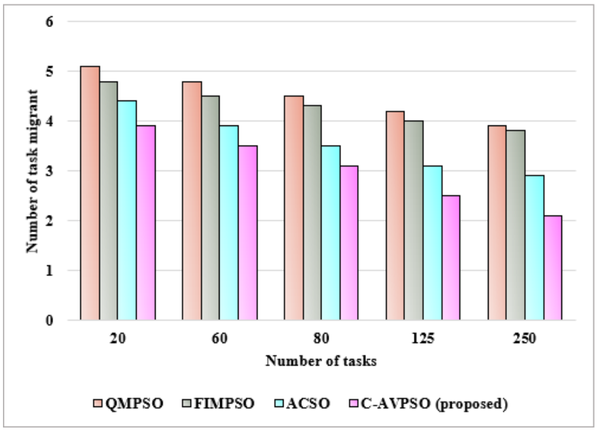

Figure 5. Migration Vs Number of tasks

Figure. 5 illustrates how the number of tasks migrated in relation via total number of tasks. Comparing the C-AVPSO technique to the current QMPSO, FIMPSO, and ACSO algorithms, it was discovered that there was less task migration.

4.1.3. Degree of imbalance

The reduction of imbalance associated with greater load balancing results in the cloud's optimal load balancing. The amount of imbalance determines how long jobs must wait. In general, the load balancing is based on how many jobs the users have requested. The degree of im-balancing after load balancing and before load balancing is depicted in Fig. 7. After the load balancing procedure, it shows that the produced C-AVPSO provide lower degree of imbalance.

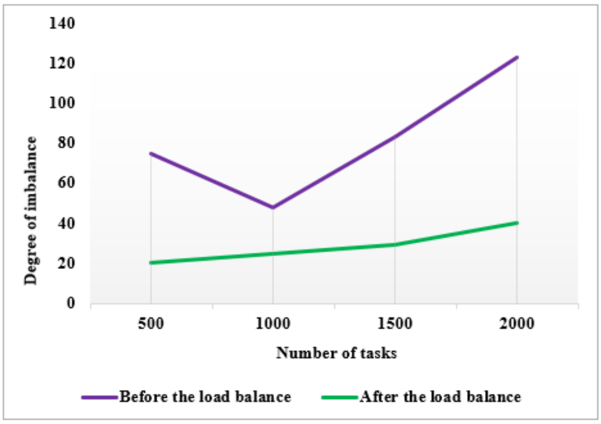

Figure 7. Degree of imbalance Vs Number of tasks

4.1.4. Resource utilization

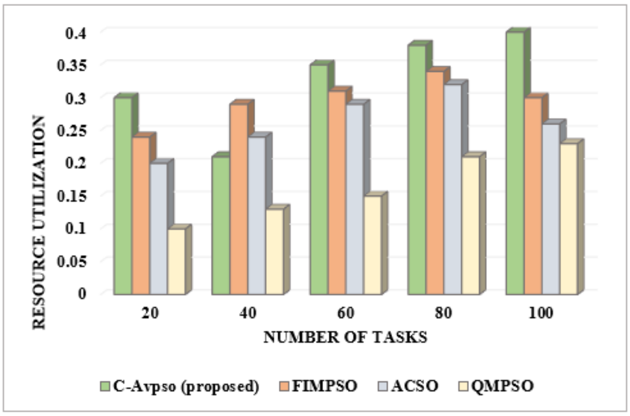

Figure 8: Resource utilization Vs Number of tasks

According to Figure 8, the suggested strategy's average resource utilisation performs admirably in every case when compared to other alternatives. This is because the suggested strategy enables the simultaneous assignment of each individual work to the best processor available. The effectiveness of the suggested strategy can be increased by maximizing resource use.

4.1.5. Response time

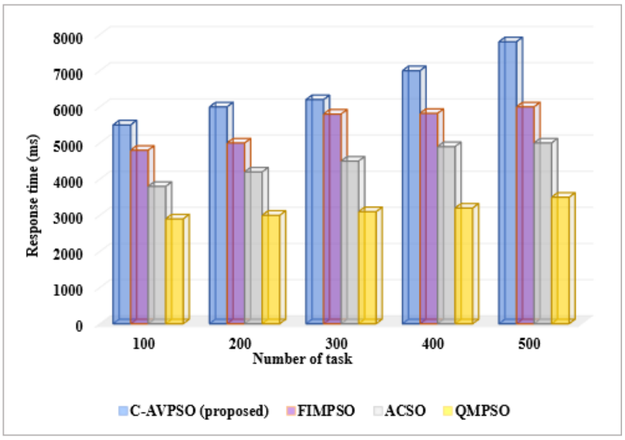

Figure 9. Response time Vs Number of task

Figure 9 shows a comparison of response times for various job counts. Between 100 to 500 jobs can be found in both the proposed and current algorithms. When a load balancing system allocates VMs with lower load conditions in response to user demand, this is known as its response time. Comparing the proposed AVPSO to the current QMPSO, FIMPSO, and ACSO approaches, it has the highest response.

5. CONCLUSION

In this paper, a novel cloud-based African vulture particle swarm optimization [C-AVPSO] has been suggested. Using C-AVPSO, the developed optimization algorithm solves the dynamic load balancing problem effectively. This approach used the AVO process to acquire the exploration space, and the PSO procedure to identify the improved response. The task's cost limitations, reaction time, and resource utilization are all satisfactorily resolved by this approach. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, in the cloud context, load balancing performance measures and convergence rate are enhanced. To enhance the operation's efficiency, the proposed method balances VM loads efficiently. The suggested method was implemented in cloud sim tool. The proposed framework is compared to existing approaches like QMPSO, FIMPSO and ACSO Based on energy utilization, degree of imbalance and task migration, resource utilization, and response time. The proposed C-AVPSO technique reduces resource utilization of 19.1%, 31%, and 54% than, QMPSO, FIMPSO and ACSO existing techniques.

REFERENCES

[1] S. K. Mishra, B. Sahoo and P. P. Parida, "Load balancing in cloud computing: a big picture", *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp.149-158, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] S. Afzal and G. Kavitha, "Load balancing in cloud computing–A hierarchical taxonomical classification", *Journal of Cloud Computing*, vol. 8, no. 1, pp.22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] V. N. Volkova, L. V. Chemenkaya, E. N. Desyatirikova, M. Hajali, A. Khodar and A. Osama, "Load balancing in cloud computing", *In 2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus)*, pp. 387-390, 2018. IEEE. [CrossRef] [Google Scholar] [Publisher Link]

[4] U. K. Jena, P. K. Das and M. R. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment", *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp.2332-2342, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] V. Polepally and K. Shahu Chatrapati, "Dragonfly optimization and constraint measure-based load balancing in cloud computing", *Cluster Computing*, vol. 22, no. Suppl 1, pp.1099-1111, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] T. P. Latchoumi and L. Parthiban, "Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment", *Wireless Personal Communications*, vol. 122, no. 3, pp.2639-2656, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for

energy efficient load balancing in cloud computing environments", *Journal of Parallel and Distributed Computing*, vol. 142, pp.36-45, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[8] A. Semmoud, M. Hakem, B. Benmammar and J. C. Charr, "Load balancing in cloud computing environments based on adaptive starvation threshold". *Concurrency and Computation: Practice and Experience*, vol. 32, no. 11, pp. e5652, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[9] K. Balaji, P. S. Kiran and M. S. Kumar, WITHDRAWN: An energy efficient load balancing on cloud computing using adaptive cat swarm optimization, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[10] M. Kumar and S. C. Sharma, "Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing", *Procedia computer science*, vol. 115, pp. 322-329, 2017. [CrossRef] [Google Scholar] [Publisher Link]
[11] S. Afzal, G. Kavitha, "Optimization of task migration cost in infrastructure cloud computing using IMDLB algorithm", *In: 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pp 1–6, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[12] R. Achar, P. S. Thilagam, N. Soans, P. V. Vikyath, S. Rao, A. M. Vijeth, "Load balancing in cloud based on live migration of virtual machines", *In: 2013 annual IEEE India Conference (INDICON)*, pp 1–5, 2013. [CrossRef] [Google Scholar] [Publisher Link]
[13] D. Magalhães, R. N. Calheiros, R. Buyya, D. G. Gomes, Workload modeling for resource usage analysis and simulation in cloud computing. Comp Elect Eng, vol. 47, pp. 69–81, 2015. [CrossRef] [Google Scholar] [Publisher Link]
[14] R. N. Calheiros, R. Ranjan, C. A. De Rose and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services", arXiv preprint arXiv:0903.2525, 2009. [CrossRef] [Google Scholar] [Publisher Link]
[15] S. Dam, G. Mandal, K. Dasgupta, P. Dutta, Genetic algorithm and gravitational. emulation-based hybrid loads balancing strategy in cloud computing, *In: Proceedings of the 2015 third international conference on computer, communication, control and information technology (C3IT)*, pp 1–7, 2015. [CrossRef] [Google Scholar] [Publisher Link]
[16] A. Dave, B. Patel, G. Bhatt, "Load balancing in cloud computing using optimization techniques: a study", *In: International Conference on Communication and Electronics Systems (ICCES)*, pp 1–6, 2016. [CrossRef] [Google Scholar] [Publisher Link]
[17] D. A. Shafiq, N. Z. Jhanjhi and A. Abdullah, "Load balancing techniques in cloud computing environment: A review", *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp.3910-3933, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[18] J. C. Bansal, "Particle swarm optimization", *Evolutionary and swarm intelligence algorithms*, pp.11-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[19] D. Wang, D. Tan and L. Liu, "Particle swarm optimization algorithm: an overview", *Soft computing*, vol. 22, pp.387-408, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[20] M. Farid, R. Latip, M. Hussin and N. A. W. Abdul Hamid, "A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing", *Symmetry*, vol. 12, no. 4, pp.551, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[21] D. Wu, "Cloud computing task scheduling policy based on improved particle swarm optimization", *In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 99-101, 2018. IEEE. [CrossRef] [Google Scholar] [Publisher Link]
[22] Y. Wang, S. Li, H. Sun, C. Huang and N. Youssefi, "The utilization of adaptive African vulture optimizer for optimal parameter identification of SOFC", *Energy Reports*, vol. 8, pp.551-560, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[23] B. Abdollahzadeh, F. S. Gharehchopogh and S. Mirjalili, "African vulture's optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems", *Computers and Industrial Engineering*, vol. 158, pp.107408, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[24] I. M. Ibrahim, "Task scheduling algorithms in cloud computing: A review", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 4, pp.1041-1053, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[25] E. H. Houssein, A. G. Gad, Y. M. Wazery and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends", *Swarm and Evolutionary Computation*, vol. 62, pp.100841, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[26] M. Masdari and M. Zangakani, "Green cloud computing using proactive virtual machine placement: challenges and issues", *Journal of Grid Computing*, vol. 18, no. 4, pp.727-759, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[27] P. Srivastava and R. Khan, "A review paper on cloud computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 6, pp.17-20, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[28] M. I. Malik, S. H. Wani and A. Rashid, "CLOUD COMPUTING-TECHNOLOGIES". *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[29] K. D. Patel and T. M. Bhalodi, "An efficient dynamic load balancing algorithm for virtual machine in cloud computing", *In 2019 International conference on intelligent computing and control systems (ICCS)*, IEEE. pp. 145-150, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[30] A. Rashid and A. Chaturvedi, A, "Cloud computing characteristics and services: a brief review", *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421-426, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[31] "Synthetic structure of industrial plastics (Journal Article)", *Plastics*, vol. 3, no. 2, pp. 15–64, 1964. [CrossRef] [Google Scholar] [Publisher Link]
[32] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, 15–64. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Dharavath Champla, currently doing as a Research Scholar, Department of Computer Science and Engineering, Anna University, Chennai, SRM Easwari Engineering college, Chennai. I have experience of over 4 years in the field of engineering, academics, administration and active research. I am an alumnus of JNTU Hyderabad. I was completed my previous Degrees, "B tech" Vidhya Bharathi Institute of Technology in 2011, "M tech" Ashoka Institute of Engineering and Technology in 2014, Affiliated to JNTU Hyderabad.

V. Ramkumar has completed his Ph.D. degree in Information and communication engineering from Anna University, Chennai, Tamilnadu. He has received his M.E degree in VLSI Design from Anna University and B.E degree in Electronics and Communication Engineering. His area of research includes VLSI Design and Optical communication. He has over 8 years of teaching experience in various reputed institutions in Tamil Nadu. Currently he is working as Assistant Professor in K. Ramakrishnan College of Technology, Tiruchirappalli. He has completed various courses on latest technologies in NPTEL, Coursera. He has delivered lectures to institutions on topics like Recent Trends in VLSI, Intern in core field, Career opportunities through GATE, Believe in Yourself, etc. He has organize & conduct various workshop and value-added course like VLSI-Cadence, IOT, PCB Design, Low Power VLSI etc., He has been scientific core committee member for International Conferences conducted in 2021.

P. Ajay is a renowned researcher and academician in the field of Wireless Networks, Artificial Intelligence and Soft Computing. He was born and brought up in India, and his passion for engineering led him to pursue a Master's degree in Electronics and Communication Engineering from Anna University, followed by a PhD degree from Anna University in 2023. Dr. Ajay began his professional journey in 2017 as an Assistant Professor at Karpagam College of Engineering, where he worked tirelessly for 3 years. During his tenure, he held various positions and made significant contributions to the institution. His technical expertise and research interests led him to explore new avenues in the field of Communication Systems and Automation, where he made significant contributions. Dr. Ajay's research work has been recognized and appreciated by the academic community, and he has received several awards and honors throughout his career. Dr. Ajay has also authored or co-authored over 25-refereed publications in journals and conferences, and he has applied for five patents that have been published in the Indian Patent Journal. Dr. Ajay is a member of various professional bodies like IEEE, MAENG, IACSIT, ISTE, and IETE. He is a reviewer for different reputed journals like Elsevier, Wiley, Inderscience, and has been a Guest Editor for a few special issues in Hindawi, Elsevier, Inderscience, Springer, etc. In his current role, Dr. Ajay is a part of the faculty at Rathinam Group of Institutions in Coimbatore, India, where he serves as a guide and mentor to students pursuing their undergraduate and postgraduate studies in the field of Electronics and Communication Engineering. He is known for his dedication, passion, and commitment to teaching and research, and his contributions have helped shape the careers of numerous students.

Arrived: 18.08.2023
Accepted: 23.10.2023