International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

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

D. Champla[1, *], Ghazanfar Ali Safdar[2], B. Muthukumar[3] and M. Mohamed Sithik[4]

[1] Department of Computer Science and Engineering, Anna University, Chennai Guindy Mambalam, Tamil Nadu 600025 India.

[2] School of Computing and Information Technology, REVA University, Bengaluru, Karnataka 560064 India.

[3] Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Sathak Nagar, SH 49, Keelakarai, Tamil Nadu, 623806, India.

[4] Department of Computer Networking, University of Bedfordshire, University of Bedfordshire, United Kingdom.

Corresponding e-mail: champla.CD32@outlook.com

Abstract – Cloud computing is a novel technology that allows consumers to access services from anywhere, at any time, under different conditions, and is controlled by a third-party cloud 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 component 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. In this method, the exploration space was obtained by using the AVO procedure whereas the enhanced response was identified by the PSO procedure. This algorithm successfully resolves resource utilization, response time, and cost constraints of the task. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, the convergence rate and performance metrics for load balancing in the cloud environment are improved. To improve 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 *– Cloud computing, Load balancing, C-AVPSO, virtual machine, Optimisation.*

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 various cloud providers available, including Amazon Web Services, Google Cloud, and others. Cloud computing is a computer model that uses the Internet to gather resources [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 from the Cloud Service Provider (CSP) [30]. With the availability of virtual cloud resources, 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].

Redistributing workloads as part of a distributed system such as the cloud ensures that no computer is overloaded or underloaded [12,13]. The technique of load balancing has assisted networks and resources in delivering the highest throughput with the quickest reaction times. [14] In 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 [17-19].

When the single VM is overloaded with tasks and there are a number of unoccupied VMs in the cloud network, it would be optimal to transfer the tasks from the overloaded

VMs to the underloaded ones [26]. Calculating every conceivable task-resource mapping in a cloud context is challenging, and finding the best mapping is not an easy process [20-24]. 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 method, the exploration space was obtained using the AVO procedure, while the enhanced response was recognized using the PSO procedure.
- 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 summarises the pertinent work in these areas with a particular emphasis on African vulture particle swarm optimisation (AVPSO) for dynamic load balancing. In this part, we've talked about a few of those technique.

In 2020 Mishra, et al., [1] proposed a category of cloud load balancing algorithms. Distinct ways of load balancing in various cloud computing systems are also described. Load balancing is the process of identifying underloaded and overloaded nodes and the balancing load between them. The simulation is performed in clouds simulator to examine the Heuristic-based performance methods, and the results are detailed.

In 2019 Afzal and Kavitha, [2] proposed a comprehensive encyclopaedic analysis about the load balancing techniques. The benefits and the disadvantages of the present techniques are outlined, and significant issues in developing effective load balancing algorithms are addressed. As a result, 80% of works do not analyze how the load balancing algorithm performs while evaluating performance.

In 2018 Volkova, 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, 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 by distributing the load among the VMs. The algorithm's resilience was demonstrated by comparing the QMPSO simulation results to the current load balancing and scheduling technique.

In 2019 Polepally and Shahu Chatrapati [5] proposed a load-balancing technique based on constraint measure. Each virtual machine's capacity and load are first computed. 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, and Parthiban, [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, [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, the highest CPU utilization of 98%, the highest memory utilization of 93%, the highest reliability of 67%, the highest throughput of 72%, and the highest make span of 148, outperforming all other compared methods.

In 2020 Semmoud, et al., [8] proposed a fresh method of load balancing for cloud computing settings. The recommended approach aims to increase system stability while lowering Makespan and VM idle time. 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 in terms of average idle time and the quantity of migrations, the HBB-LB algorithm.

In 2021 Balaji, et al., [9] proposed, a load balancing system that addresses optimisation problems by using the adaptive cat swarm optimisation (ACSO) technique. The effectiveness of the suggested technique 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, and Sharma, [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 experimental findings demonstrate that the suggested technique decreases 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 takes into account the cost, response time, and resource utilization in order to optimize dynamic load balancing. Using this method, you can increase the throughput of your virtual machines, distribute the load among the virtual machines, and maintain the balance of task preferences by adjusting the waiting times for complicated tasks.

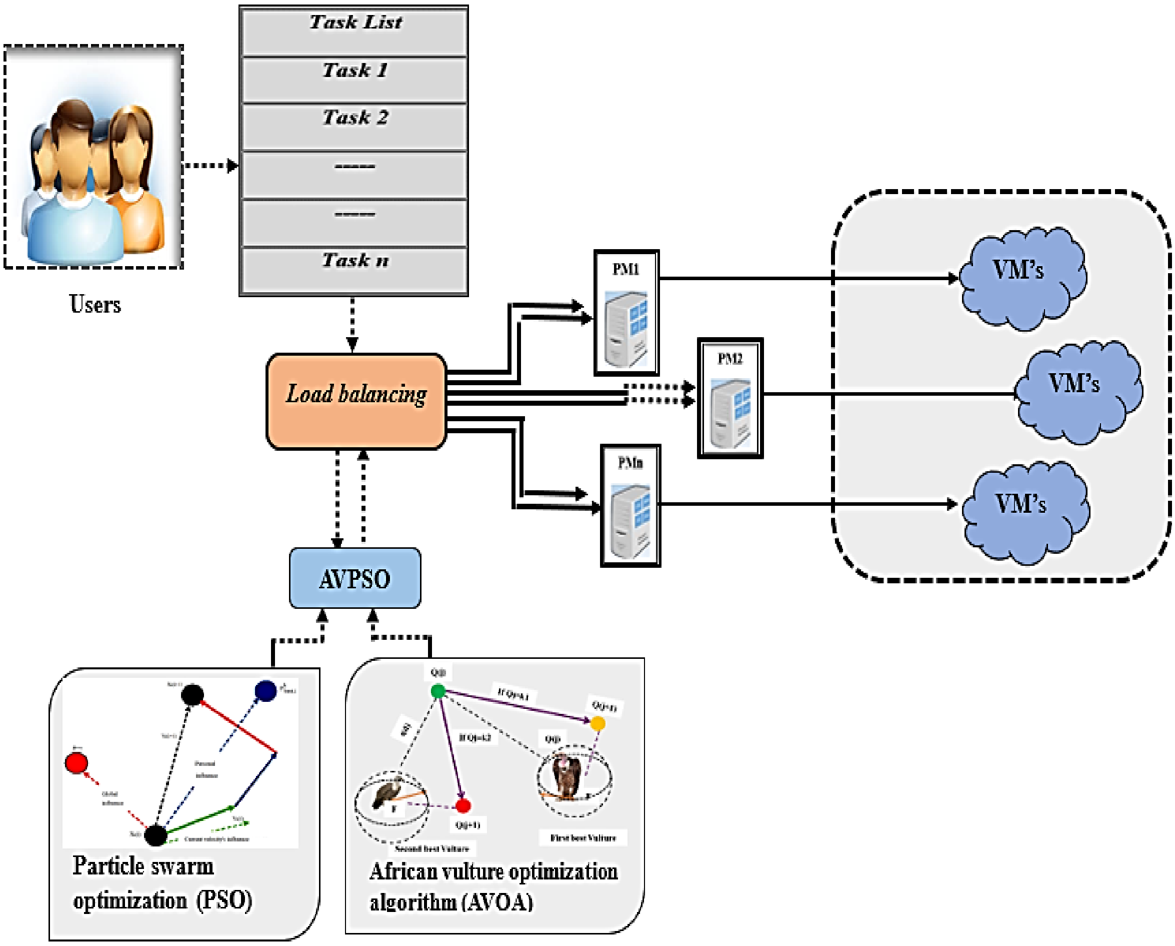

Figure 1. Block diagram of proposed methodology

Fig.1 illustrates the proposed methodology. The primary process in the cloud is workload distribution to virtual machines (VMs). Using node performance data, load balancing decisions are made 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 over tasks from the overloaded VMs. Due to this, the performance and the latent times are sped up. By effectively distributing the load in the cloud, the suggested work-based load balancing system increases resource utilisation 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 from the worst overloaded VMs to the superior underloaded VM. 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 depicts the scheduling task in cloud. Each procedure is carried out in the cloud surroundings thanks to cloud computing, which offers cloud facilities to cloud users. Due to the enormous volume of diverse input tasks so as to balance the demands of diverse resources, load balancing is necessary.

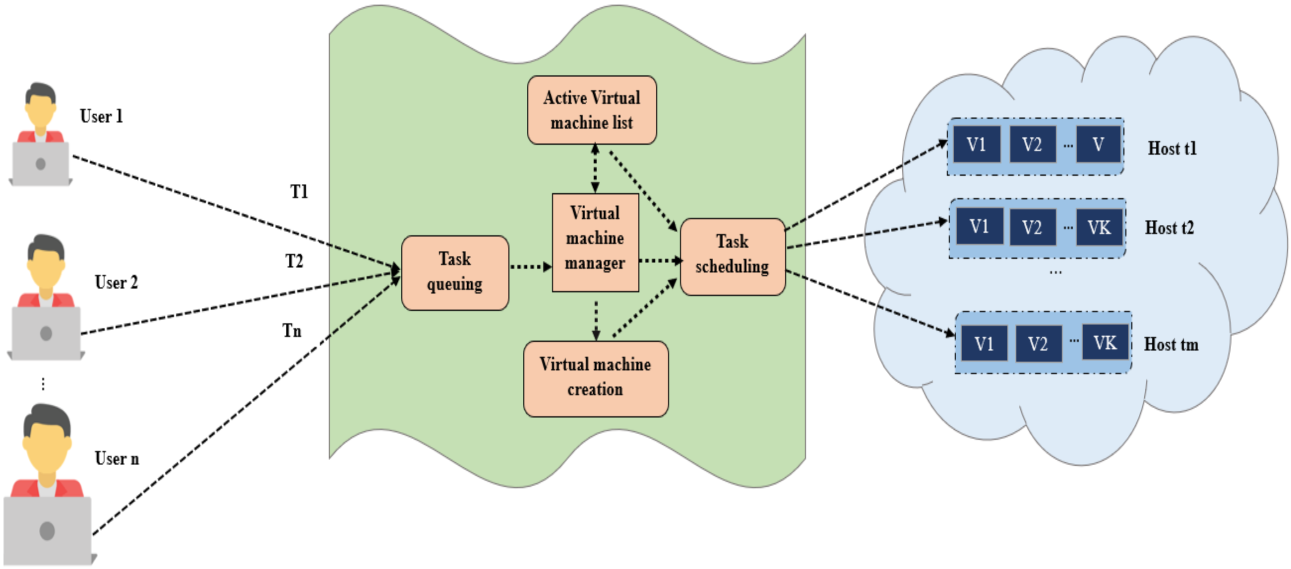

Figure 2. Scheduling cloud tasks for load balancing

The cloud system's task queue receives the tasks with n inputs T_(1),T_2,…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 can only support a certain number of active VMs.

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 problem space. It differs from traditional optimisation techniques 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., $x_i^t = (x_{i1}^t, x_{i2}^t, \ldots . x_{iD}^t)^T$. Likewise, the velocity at step time t can be represented by another D−dimensional vector $v_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t)^T$.The earlier position of the i^{th} particle at the step time t is denoted as $p_i^t = (p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t)^T$. $'g'$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:

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

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

Position update equation:

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

Where, $d = 1, 2, \ldots D$ denotes the dimensions and the particle index is represented by $i = 1,2, \ldots s$. S is the swarm's size, whereas c1 and c2, or the cognitive and social scaling parameters, are constants. Equations (1) and (2) seem to Indicate that the dimensions of each particle are updated independently. 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**

Apply the velocity update equation 1;

Apply position update equation 2;

end

Compute fitness of updated position;

If needed, update historical information for pbest and gbest;

end

Terminate if gbest meets problem requirements;

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 unique vultures that can soar to the highest point among the many vulture species. The rotational movements of the African vultures in the sky cause them to fly constantly from one location to another in search of better food supplies. 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.

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

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,

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

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

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

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

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

4. RESULT AND DISCUSSION

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

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). Energy utilization vs number of VM's and number of tasks is shown in Figures 3 and 4.

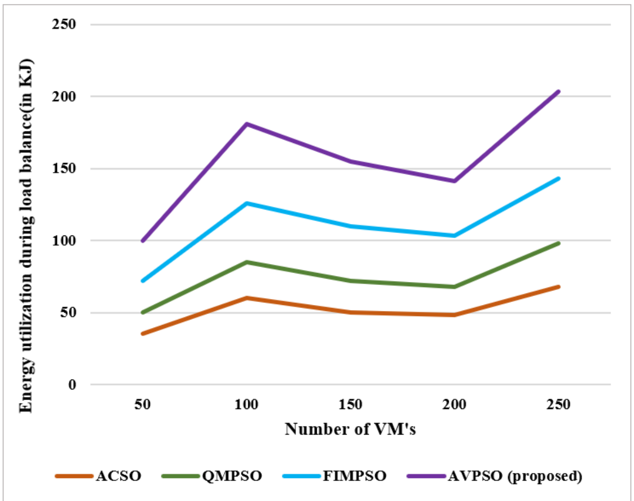

Figure 3. Energy utilization vs number of VM's

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.

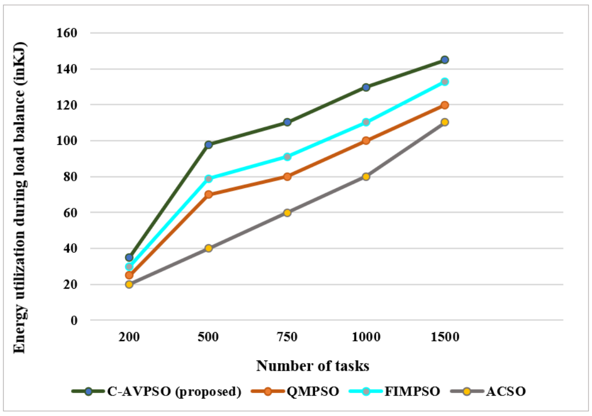

Figure 4. Energy utilization vs number of tasks

4.1.2. Migration

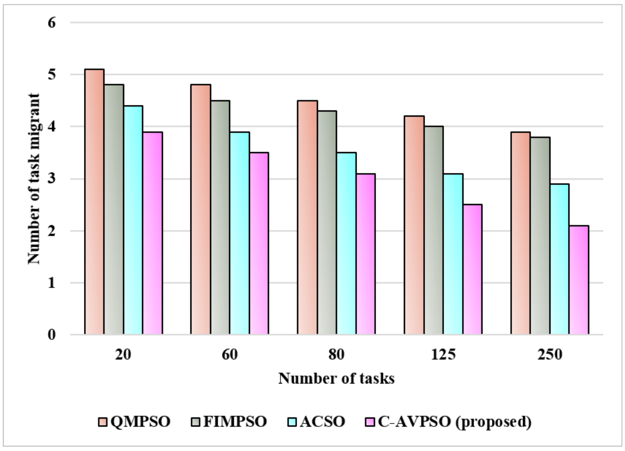

Figure 5. Migration Vs Number of tasks

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

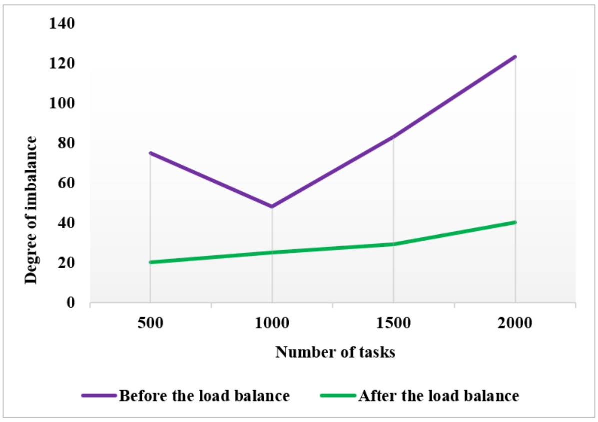

Figure 6. Degree of imbalance Vs Number of tasks

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. 6. After the load balancing procedure, it shows that the produced C-AVPSO provide lower degree of imbalance.

4.1.4. Resource utilization

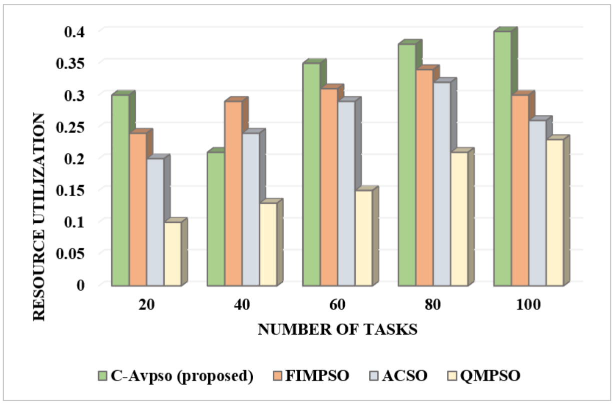

Figure 7. Resource utilization Vs Number of tasks

According to Figure 7, 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 maximising resource use.

4.1.5. Response time

Figure 8 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.

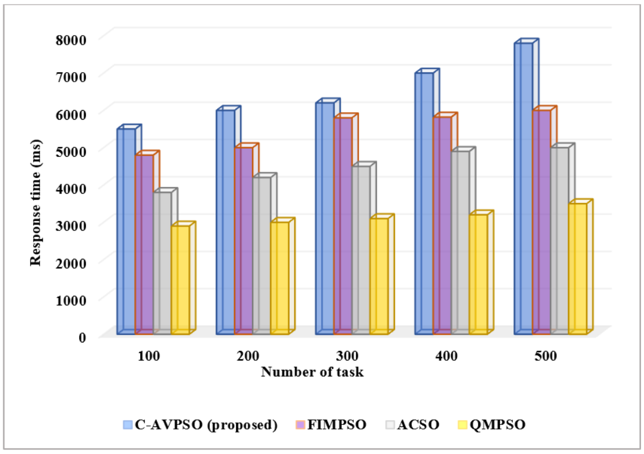

Figure 8. Response time Vs Number of task

5. CONCLUSION

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. In this method, the exploration space was obtained by using the AVO procedure whereas the enhanced response was identified by the PSO procedure. This algorithm successfully resolves resource utilization, response time, and cost constraints of the task. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, the convergence rate and performance metrics for load balancing in the cloud environment are improved. To improve 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, 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.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

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)*, IEEE. pp. 387-390, 2018. [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*, 22(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, 2020. [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, and 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, December). [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, December). [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, February. [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, October. [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. [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 & 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. Bhalodia, "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, "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]

AUTHORS

D. 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.

Ghazanfar Ali Safdar received a BSc (Hons) in Electrical Engineering from University of Engineering and Technology, Pakistan in 1997. He subsequently received M. Eng in Computer Science and Telecommunications from University of Technology, Grenoble, France in 2001.He then worked from 2005-2008 with the same university as a Research Fellow involved in the security and authentication protocols for wireless networks. Ghazanfar Ali is currently working as a lecturer in Computer Networking at University of Bedfordshire. He has also worked as R & D engineer with Carrier Telephone Industries (SIEMENS), Pakistan and Schlumberger, France. Dr Ghazanfar Ali Safdar, has been selected to be an associate editor for IET Networks, the journal of the prestigious Institution of Engineering & Technology.

B. Muthu Kumar, Professor, School of Computing and Information Technology, REVA University, Bengaluru received his B. E (CSE) degree from Anna University, Chennai in the year 2005, M.Tech (CSE) (Gold Medalist) received from Dr. MGR. University, Chennai in the year 2007 and Doctoral degree from St. Peter's University, Chennai in the year 2013. He is having more than 16 years of teaching experience in reputed engineering colleges. He has published more than 40 peer reviewed International Journals, 50 International/National Conference and attended more than 150 Workshops/FDPs/Seminars etc., He organized many events like Conference/FDPs/Workshops/Seminars/Guest Lecture. He has published more than 10 patents in various fields like Wireless Sensor Networking, Image Processing, Optimization Techniques and IoT. He received nearly 5.67 Lakhs funding from various agencies like AICTE, ATAL and IEI. He has written 2 books from reputed publishers. He received Best Researcher Award in the year 2021 and Innovative Research and Dedicated Professor Award in Computer Science and Engineering in the year 2018. He has professional membership on ISTE, CSI, IEI, IACSIT, IAENG, CSTA, and SIAM. He has invited as Guest Lecture / Chairperson / Examiner / Reviewer/Editorial Board Member in various Institutions/Journals/Conferences. He is a recognized supervisor in Anna University, Chennai and currently guiding 4 research scholars. His areas of interest are Image Processing, Wireless Networks, IOT and Computing Techniques.

M. Mohamed Sithik, Associate Professor, Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Kilakarai received his B.E (CSE) degree from Madurai Kamarajar University in the year 2004, and M.E CSE received from Anna University Tiruchirappalli in the year of 2011. He is having more than 18 years of teaching experience in reputed engineering college. He has published more than 5 peer reviewed Journals, 10 International/National Conference and attended more than 100 Workshops/FDPs/Seminars etc., He organized many events like Conference/FDPs/Workshops/Seminars. He has published 2 patents in the area of Artificial Intelligence and Telehealth Technologies. He has written 1 book from reputed publisher. He received Best Techie Award in the year 2022. He has professional membership on IEEE. His current research focuses on Secure RPL and Congestion control in Internet of Things (IoT). His research interests include Data Science, Deep Learning, Load balancing, Energy consumption, Security in IoT devices.

Arrived: 08.11.2023

Accepted: 11.12.2023