Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

# Load Balancing in Cloud Computing Using Meta-Heuristic Algorithm: Survey

Aliaa Ayoob Hadi[1,*], Huda Hallawi [2]

Department Of Computer Science, University of Karbala, Karbala,56001, Iraq
aliaa.a@s.uokerbala.edu.iq
huda.f@uokerbala.edu.iq
[*]Corresponding author E-mail: aliaa.a@s.uokerbala.edu.iq

https://doi.org/10.46649/fjiece.v3.2.29a.8.6.2024

***Abstract.*** *Load balancing is essential within the cloud. Distributing the workload over many servers permits for a more efficient machine. Optimal resource use, decreased time to response, and more advantageous gadget performance are all results. Among the numerous challenges that traditional methods of accomplishing equilibrium come upon are problems with adaptability and the unpredictability of cloud structures. This survey focuses on the demanding situations associated with load balancing in cloud computing and proposes the usage of specialized algorithms as an answer. It determines the most reliable alternatives for facts placement in the cloud. Meta-heuristic algorithms draw concept from herbal approaches, which includes the foraging behavior of ants or the hunting strategies of predators. These algorithms excel at rapidly finding super answers in situations where conventional optimization algorithms fail to get great results. Various strategies have been proposed to attain paintings equilibrium in cloud computing. Evidence has shown that these techniques can enhance work balance in cloud computing.*

***Keywords:*** *Meta -Heuristic Algorithm; Cloud Computing; Load Balancing Algorithms.*

## 1. INTRODUCTION

Load balancing is critical in cloud computing to distribute the workload frivolously amongst all nodes, preventing eventualities in which some nodes are overloaded even as others are idle or underutilized. Load balancing ensures gold standard resource usage through maximizing use available machine assets, avoiding aid wastage. It enables reaching higher overall performance by lowering response time and improving throughput, because the workload is lightly disbursed throughout the cloud infrastructure. Load balancing enhances scalability and elasticity, permitting the machine to handle growing workloads and dynamically adapt to changing needs. Efficient load balancing algorithms and strategies make a contribution to improved fault tolerance and reliability, as they could hit upon and redirect visitors from overloaded or failed nodes to healthful ones [1][2]. For complicated responsibilities, metaheuristic algorithms are optimization techniques. Natural phenomena like as evolution and genetics function inspiration [3]. The categorization and the significance of metaheuristic algorithms covered in this study. Gives a list of well-known metaheuristic algorithms, such as The Honey Bee Foraging

Algorithm, Particle Swarm Optimization, Genetic Algorithms, and Ant Colony Optimization. The scope of the paper discusses different metaheuristic load balancing methods. Cloud computing has increased demand for online resources. Load balancing algorithms are needed to manage traffic[1]. A task scheduling technique with various objectives aims to reduce scheduling time and energy consumption. The Whale Optimization Algorithm optimizes task scheduling by considering voltage frequency, execution time, and job sequence. Simulated using MATLAB, the workload is generated randomly and compared to the conventional PSO method, significantly reducing energy usage and execution time[2].The authors proposed a scheduling method that takes into account both makespan and processing costs. A two-step scheduling approach was designed. During the first step, technique of order precedence by similarity to ideal solution (TOPSIS) was utilized to identify optimum jobs while considering restrictions. In the second step, PSO was employed to optimize job scheduling on virtual machines (VMs). The suggested technique reduces makespan and processing costs while increasing resource usage as compared to other PSO versions[3]. a technique for scheduling tasks that maximizes a number of important metrics, such as the use of cloud resources and the makespan time of expected workloads. Enhanced Multiverse Optimizer (MVO/PSO) is a methodology for solving task scheduling problems; it combines the two algorithms to prevent PSO's premature convergence problem. The authors utilized a synthetic dataset for task creation and executed their simulations on Cloudsim. It compared to preexisting MVO and PSO, which optimized resource use while minimizing makespan and throughput[4]. The job scheduling algorithm was developed using a modified ant colony optimization approach. The writers focus solely on Makespan. This method divides tasks into sub lists and schedules them for appropriate virtual resources. Cloudsim is used for simulation. It was likened to ACO. The simulation results revealed a reduction in the makespan compared to ACO[5]. The authors recommended a multi-goal work scheduling technique that addresses both makespan and power consumption. A changed genetic algorithm was hired as the technique for fixing the work scheduling problem. Cloudsim become used for simulation and compared to present day strategies such as min-min, satisfactory in shape, hill mountaineering, and random search algorithms. Results confirmed that the suggested method outperformed existing techniques by decreasing makespan and energy usage [9]. A scheduling technique was created to do not forget factors inclusive of makespan and energy intake. A hybrid variant of GA was utilized to calculate priorities based totally at the produced chromosomes, resulting in a truthful scheduling mechanism for cloud computing. The implementation used Matlab 2014 and turned into evaluated in opposition to present PSO and GA versions. Simulation findings showed tremendous discounts in energy usage and makespan [10]. A scheduling method changed into advanced to manage electricity intake and makespan. The undertaking scheduling set of rules utilizes a Non-dominated Sorting Genetic Algorithm (NSGA-II) technique. The created population is split into groupings of same or special sizes based on their precedence. It changed into tested in opposition to numerous GA techniques. Moreover, the proposed method reduced power utilization and makespan [11]. A task scheduling strategy become suggested that considers makespan and strength utilization. The "CSSA (chaotic squirrel search set of rules)" technique changed into utilized to address the task scheduling trouble. Cloudsim become used to simulate and compare hybrid GA-PSO and BAT algorithms, ensuing in large reductions in makespan and electricity usage [12]. A scheduling strategy turned into provided to minimize strength consumption by optimizing the usage of virtual machines (VMs). The BWM and TOPSIS algorithms have been hired to schedule jobs efficaciously. The great-worst method (BWM) algorithm changed into used to assign weights, at the same time as the TOPSIS set of rules was used to assign priority scores. A strength dispatcher was then hired to decrease energy utilization and decorate VM utilization. Simulation become completed the use of the Cloudsim platform. It turned into evaluated the usage of the Dynamic voltage and frequency scaling (DVFS) technique. Ultimately, the advised technique has appreciably inspired the electricity consumption and use of virtual machines (VMs) [13]. A power-green task scheduling machine has been devised. This mechanism schedules jobs in two tiers, taking into consideration the varying workloads of different sorts.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

The scheduling manner employs a genetic algorithm as a methodology. The implementation is divided into elements. The initial phase involves mapping work onto digital assets without contemplating any cut-off date constraints. In the second one section, obligations are assigned to digital machines (VMs) based on a mission reassignment policy that takes into account undertaking precedence. It is implemented on the Amazon Cloud platform. When in comparison to baseline techniques which include Shortest Job First (SJF), First Come, First Served (FCFS), and Genetic set of rules (GA) variations, the new strategy turned into proven to efficaciously cut electricity use and satisfy activity cut-off dates with precision [14]. A scheduling system was created by combining opposition-based gaining knowledge of and particle swarm optimization (PSO) algorithms. Opposition-primarily based mastering turned into hired to circumvent the nearby most useful and mitigate the drawbacks of the Particle Swarm Optimization (PSO) set of rules. The standards addressed include makespan, PIR ratio, and diploma of unbalance. Cloudsim become utilized as a simulator to behavior a simulation and changed into then as compared to different modern-day methodologies together with PSO, mPSO, GA, Min-Min, and Max-Min. The advised approach verified good sized improvements inside the distinctive parameters as compared to present procedures, as seen via the simulation consequences [15]. An undertaking scheduling gadget became created to specifically address the parameter known as makespan. This method takes into account the workload from different heterogeneous assets. The Crow search set of rules changed into employed as a methodology that dynamically adjusts its flight duration, making it properly-desirable for paintings scheduling mechanisms in cloud computing. Cloudsim changed into utilized for simulation and contrasted in opposition to algorithms such as Max-Min, Min-Min, and a counselled method. The proposed approach considerably reduced the makespan as compared to current algorithms [16]. Discusses The task scheduling approach considers both processes finishing touch time and resource utilization. This method utilized a normalization manner for green process scheduling. The implementation was MATLAB-primarily based, with randomly generated jobs that addressed factors like makespan and power intake. They utilized artificial datasets to validate the randomly generated findings. Compared to Round Robin and MaxUtil, the proposed method significantly reduces energy intake and makespan [17]. The authors created a work scheduling system that reduces makespan and operating expenses. A hybrid technique combining oppositional based learning (OBL) and cuckoo search algorithm (CSA) algorithms was employed to tackle the work scheduling problem. Simulation utilizing Cloudsim was compared to PSO, Improved Differential Evolution Algorithm (IDEA), and GA. Th recommended technique improved both time and service costs[6].The authors proposed a scheduling technique for cloud computing using the "Locust Inspired Scheduling Model" that considers aspects such as energy usage, response time, and processor utilization. The Cluster Computing Cloudsim simulator was implemented and compared to established benchmarks such as DVFS, Energy-aware Scheduling using the Workload-aware Consolidation Technique (ESWCT), and Threshold with Minimum Utilization (ThrMu). The results. A showed that it significantly reduced energy usage, response time, and increased processing utilization[7].The authors systemically organized the paper. It consists of four sections: Section 1 discusses the introduction; Section 2 discusses load balancing challenges and issues; Section 3 discusses the load balancing algorithm classification; and Section 4 discusses the conclusion.

Table 1 *Summary of Existing Load Balancing and Task Scheduling Techniques*

| **Authors** | **Algorithm/ Technique** | **Simulation environment** | **Metrix** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Peng et al.[2] | Whale optimization | Matlab | Execution time, energy consumption |
| Panwar et al.[3] | TOPSIS and PSO | Cloudsim | Makespan, processing cost, resource utilization |
| Shukri et al.[4] | MVO and PSO | Cloudsim | Makespan, resource utilization |
| Sharma et al.[5] | Modified ant colony | Cloudsim | Makespan |
| Vila et al.[8] | Modified GA | Cloudsim | Makespan and energy consumption |
| Pirozmand et al.[9] | Hybrid GA | Matlab | Makespan, energy consumption |
| Shukla et al.[10] | NSGA-II | Cloudsim | Makespan, energy consumption |
| Sanaj et al.[11] | CSSA | Cloudsim | Makespan, energy consumption |
| Khorsand et al.[12] | BWM and TOPSIS | Cloudsim | Energy consumption and VM utilization |
| Hussain et al.[13] | Genetic algorithm | AWS cloud environment | Energy consumption |
| Agarwal et al.[14] | Hybrid OBL and PSO | Cloudsim | Makespan, degree of imbalance and PIR Ratio |
| Prasanna Kumar et al.[15] | Crow search algorithm | Cloudsim | Makespan |
| Panda et al.[16] | Normalization procedure | Matlab | Makespan, energy consumption |
| Krishnadoss et al.[6] | OBL and CSA | Cloudsim | Energy consumption and makespan |
| Kurdi et al. [7] | Locust scheduling model | Cloudsim | Makespan, energy consumption |

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

## 2. Challenges In Load Balancing

It is important to solve all the main problems that impact the performance of load balancing algorithms. When aiming to enhance the performance of a load balancer, it is necessary to consider the following issues.

### 2.1 *Throughput*

relates to the amount of work completed during a given timeframe. It is a measure of the system's efficiency. It is one of the most important challenges in cloud computing. The studies aim to enhance throughput and reduce latency in cloud computing by prioritizing load balancing for independent work in cloud scheduling algorithms[17].

### 2.2 *Associated overhead*

the cost of loading balancing is defined as the time required to carry out the additional computations and communications – known as the overhead – necessary to implement a load-balancing algorithm. Overheads include moving tasks between nodes, communications between processes, and between processors. The overhead should be kept to a minimum for the load balance method to work properly[18].

### 2.3 *Fault tolerance*

This term refers to the capacity of a load balancing algorithm to work even when there are random link or node failures. The integration of fault tolerance on load balancing algorithms guarantees that the system is stable and can withstand failures while running without interrupting its intended operation. Failures in cloud computing may be classified into two groups. First, software vulnerabilities include information exploitation and inadequate data from the source. Secondly, malfunctioning or slow virtual machines, excluding storage access, are examples of hardware failures[19].

### 2.4 *Migration time*

This term refers to the time taken for a process to move from one node to the other system for task execution. Reducing the migration time is significant since resources are allocated efficiently as well as the task execution. A short migration time ensures that tasks are transferred between nodes effectively and quickly such that the distributed system needs to balance the load hence maximizing its resources to perform better. Various migration strategies, such as pre-copy, post-copy, adaptive compression, LRU, splay tree, checkpoint recovery trace, and replay method, can be employed for transferring a virtual machine between hosts[20].

2.5 *Response time*

refers to the time taken by the load balancing technique to respond in the distributed system. It is one of the most important challenges in cloud computing. The timer begins when a client submits a request and ends when the server provides its initial response, measured in milliseconds[21].

2.6 *Resource utilization*

It is also referred to as the resources in a cloud computing environment that measures the effectiveness and efficiency with which the cloud computing resource is used to determine the given parameter's resource's present utilization. Good load balancing technique should maximize the use of resources[22].

2.7 *Scalability*

Refers to the capability of an algorithm or system to manage an increasing number of processors and machines without a significant decrease in performance. Without the use of efficient load-balancing techniques, cloud computing cannot achieve scalability[23].

2.8 *Performance*

refers to the overall effectiveness of a system. It measures how well a system is able to perform its tasks and deliver results. Improving performance involves enhancing various parameters and factors that contribute to the system's overall effectiveness[24].

2.9 *Geographical Distributions of the Nodes*

Cloud computing is essential for large-scale applications like Facebook and Twitter and benefit from the geographical distribution of nodes. The distribution of nodes within the cloud computing environments is also a highly critical aspect that would help systems remain efficient and appropriately leverage fault tolerance[25].

2.10 *Emergence of Small Data Centers for Cloud Computing*

Small data centres are gaining popularity due to their potential benefits, such as lower costs and energy consumption compared to large data centers. These small data centers can provide cloud computing services, contributing to geo-diversity computing and improving overall performance[26].

2.11 *Stored Data Management*

The management of data storage presents a significant challenge within the realm of cloud computing, impacting both enterprises that delegate their data storage responsibilities and individual users. The volume of data housed within networks has experienced a substantial surge over the last ten years, necessitating the implementation of effective management tactics[27].

### 2.12 *Energy Management*

Cloud computing enables the sharing of a collection of global resources among fewer providers, instead of each provider having its own resources. This results in significant economies of scale. To achieve this, energy conservation is crucial as it helps in utilizing a portion of the data center while maintaining acceptable performance levels[28].

### 2.13 *Virtual Machines Migration*

Virtual machines possess the capability to be transferred among physical machines in order to alleviate the burden on heavily utilized physical machines. Virtualization concept facilitates the perception of an entire machine as a singular file or collection of files, thus streamlining the process of transferring virtual machines[29].

### 2.14 *Automated Service Provisioning*

Elasticity in cloud computing involves automatically allocating or releasing resources. The challenge is to utilize cloud resources effectively while maintaining performance levels similar to traditional systems and optimizing available resources[30].

## 3 Classification of Load Balancing Algorithms

Various load balancing methods are frequently used to improve the performance of cloud computing. These algorithms are often classified as static, dynamic, or nature-inspired based on the contexts in which they work. Additional information on these categories will be given below.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

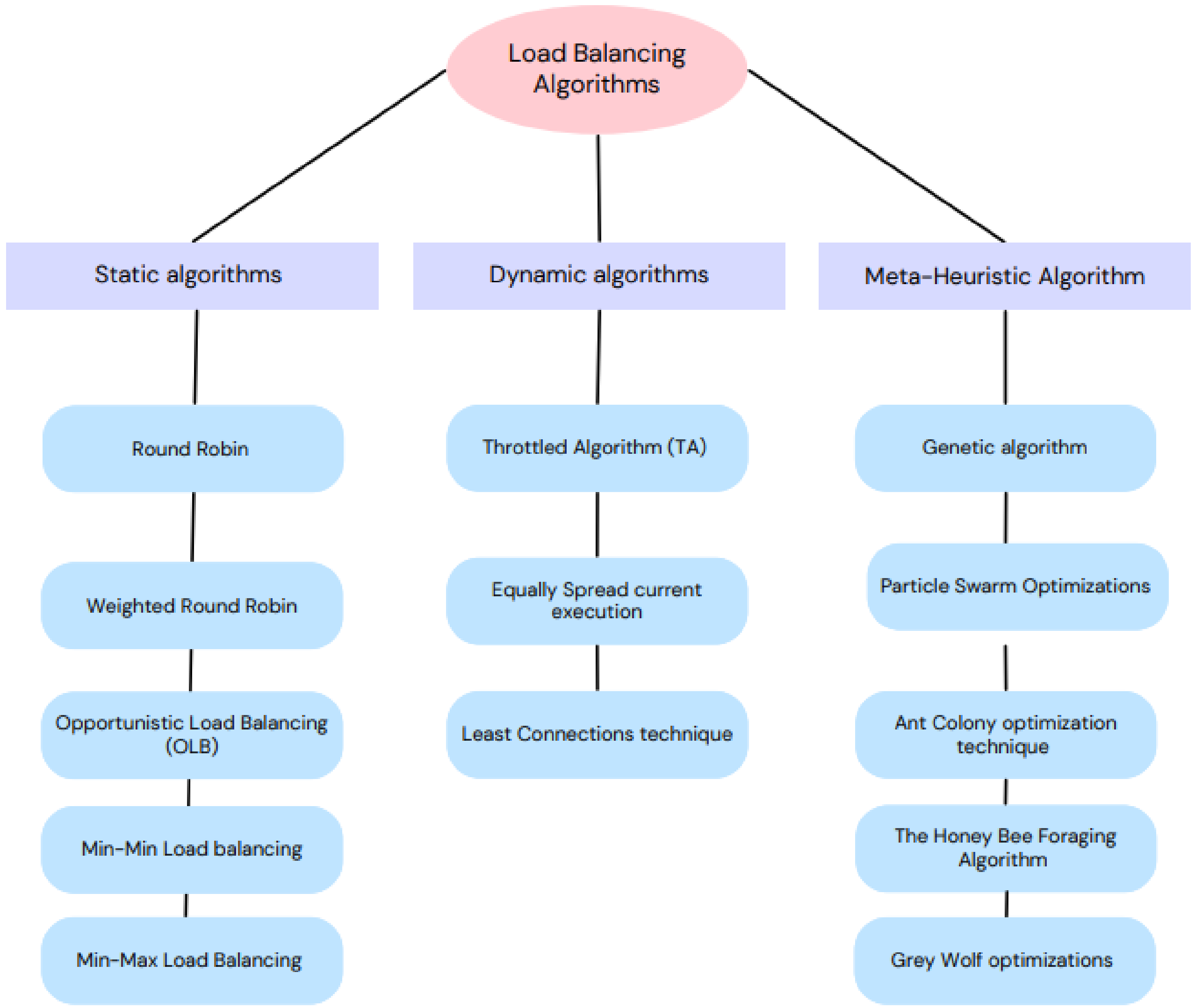


**Fig. 1. classification of Load Balancing Algorithms**

*3.1 Static algorithms*

Static algorithms make load balancing decisions at compile time based on previous knowledge of system attributes and do not require information about the current state of the system , making them less complex and suitable for systems with low load variation [31].

The following are the most commonly used static load balancing techniques:

*a- Round Robin*

The round-robin load-balancing method uses a time-triggered scheduling scheme to allocate tasks to machines, relying on data centers and randomly selecting nodes for load balancing. It is practical, reliable, and commonly used in cloud systems[32].

*b- Weighted Round Robin*

The Weighted Round Robin approach distributes a powerful virtual machine to processes with greater workloads and assigns a weight to each process depending on its capacity. At the same time, it takes longer than the round-robin technique[33].

*c- Opportunistic Load Balancing (OLB)*

OLB is a static load-balancing approach, which does not consider each device's current workflow. The OLB approach seeks to keep every server active by randomly distributing all unfinished work activities among available servers. However, this approach can lead to disappointing results in task scheduling. The OLB method struggles to determine the complexity of each node, which further decreases the efficiency of processing activities. As a result, the cloud system may experience bottlenecks, which are points of congestion or slowdown in the system[34].

*d- Min-Min Load balancing*

The approach starts via an unmapped list of tasks. Using this strategy, the machine that completes all tasks within the shortest time is chosen. When a user request has a minimum completion time requirement, it assigns resources to that request. System state and node data are recorded in a table. Until every unmapped activity is allocated to a virtual machine (VM), the allocation procedure is repeated[35].

*e- Min-Max Load Balancing*

The Min-Max Load Balancing approach aims to decrease the amount of time it takes for large operations to complete by assigning tasks to machines according to their completion times, with the lowest time allocated first and the highest amount of time distributed to an individual resource.[36]

*3.2 Dynamic algorithms*

These algorithms depend to make decisions mainly on the state of the system at that moment, they do not take past system information into account. In order to balance the load, dynamic algorithms take into account many criteria, including transfer, selection, location, and information policies. It also takes into account the nodes' dynamic state changes. A node with a high load is always switched to one with a light load.

The following are the most commonly used Dynamic load balancing techniques:

**a-** *Throttled Algorithm (TA)*

The method for dynamic load balancing looks for an appropriate virtual machine (VM) to carry out activities. When a VM becomes available and has sufficient capacity, the TA accepts the job; if not, it queues the request for quick processing. TA keeps an index table with the status of every VM. However, TA does not consider advanced load balancing requirements like Processing Time[37].

**b-** *Equally Spread current execution*

The Equally Spread current execution algorithm is designed with the objective of evenly distributing the workload among all servers. This algorithm ensures that each server executes an equal amount of workload, preventing any server from being overloaded while others remain underutilized. This approach contributes to maintaining the servers' application performance by distributing the workload equally. The workload distribution in the cloud computing environment can be further optimized by combining the Equally Spread current execution method with other load balancing strategies [37].

**c-** *Least Connections technique*

The "Least Connections" load balancing strategy transfers the load to the server with the fewest active transaction data. A dynamic scheduling strategy routes user requests to the cloud server with the fewest active connections.[36]

3.3 Meta-Heuristic algorithms

are a set of optimization algorithms that draw inspiration from processes and phenomena seen in nature. To solve difficult optimization issues, these algorithms simulate the actions of natural systems, such as ants, bees, birds, or genetic evolution. Their purpose is to efficiently and effectively explore the search space to uncover nearly ideal solutions [38].

The following are the most commonly used Meta-Heuristic algorithms load balancing techniques:

**a-** *Ant Colony optimization technique*

When real ants construct a network to look for food, they move ahead, monitoring node loads. If they find an overloaded node, they travel backward to a previously underloaded node to share data. This is the basis for the ant colony optimization technique [39].

**b-** *The Honey Bee Foraging Algorithm*

It is a decentralized load-balancing technique that draws inspiration from honey bee behaviour , which helps balance the load across different nodes in a cloud system by removing tasks from overloaded nodes and assigning them to lightly loaded nodes based on priority.[39]

**c-** *Genetic algorithm (GA***)**

It is an optimization technique based on population. that represents possible solutions as chromosomes and uses a fitness function to evaluate their suitability.

GA involves selecting chromosomes based on fitness value, producing offspring through crossover and mutation operations, and repeating the process until adequate offspring are generated. In the context of scheduling and load balancing, GA assigns genes to virtual machines for task execution [41][43].

**d-** *Particle Swarm Optimizations (PSO)*

Fish schools and bird swarming are examples of social behaviour that served as inspiration for the PSO algorithm, where a swarm of particles represents candidate solutions that search for the best solution in a given space. It is known for its simplicity, fewer controlling parameters, and flexibility to hybridize with other optimization algorithms[44].

**e-** *Grey Wolf Optimization algorithm*

The Grey Wolf Optimization (GWO) approach is a nature-inspired metaheuristic developed in 2014 to solve complicated optimization problems by imitating grey wolves' social structure and hunting methods. This technique categorizes possible solutions into four hierarchical levels: alpha, beta, delta, and omega, which correspond to the leadership structure of a wolf pack. The optimization approach is separated into three phases: surrounding prey, hunting, and attacking prey, with position updates driven by the top three solutions that mimic the wolves' cooperative behavior[45].

## 4- CONCLUSION

Load balancing is important for boosting device performance and useful resource optimization in cloud computing. Traditional strategies often war from scalability issues and fail to alter flexibly with changing operations. Or heterogeneous aid configurations. In order to manage these issues, meta-heuristic algorithms have proven to be an effective device. Meta-heuristic algorithms like particle swarm optimization and genetic algorithms offer adaptive decision abilities that may correctly deal with static and dynamic workload distributions. Research is focusing on refining present meta-heuristic algorithms or developing new approaches that can cope with the rising challenges of load balancing in cloud computing subsequently main to offerings that extra strong and powerful cloud-based totally has emerged. To summarize, the use of dynamic algorithms guarantees that the distribution of assets is adaptable to the changing conditions, which ensures powerful reaction to workloads. On the other hand, static algorithms bring the detail of balance and predictability into the machine, which ensures that workloads with fixed aid necessities may be managed within the first-rate manner. Algorithms that exist thanks to legal guidelines of nature leverage biological tactics and behaviors to cope with complex optimization issues, and such nature-stimulated answers are progressive way of improving control of cloud sources. Each of those strategies has very own strong factors, depending on precise necessities of the cloud surroundings.

## REFERENCES

[1] K. Krishna Sowjanya and S. K. Mouleeswaran, "Load Balancing Algorithms in Cloud Computing BT - Proceedings of the International Conference on Cognitive and Intelligent Computing," A. Kumar, G. Ghinea, S. Merugu, and T. Hashimoto, Eds., Singapore: Springer Nature Singapore, 2023, pp. 483–493.

[2] H. Peng, W.-S. Wen, M.-L. Tseng, and L.-L. Li, "Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment," *Appl. Soft Comput.*, vol. 80, pp. 534–545, 2019, doi: https://doi.org/10.1016/j.asoc.2019.04.027.

[3] N. Panwar, S. Negi, M. M. S. Rauthan, and K. S. Vaisla, "TOPSIS–PSO inspired non-preemptive

tasks scheduling algorithm in cloud environment," *Cluster Comput.*, vol. 22, no. 4, pp. 1379–1396, 2019, doi: 10.1007/s10586-019-02915-3.

[4] S. E. Shukri, R. Al-Sayyed, A. Hudaib, and S. Mirjalili, "Enhanced multi-verse optimizer for task scheduling in cloud computing environments," *Expert Syst. Appl.*, vol. 168, p. 114230, 2021, doi: https://doi.org/10.1016/j.eswa.2020.114230.

[5] S. Sharma and R. Jain, "EACO: An Enhanced Ant Colony Optimization Algorithm for Task Scheduling in Cloud Computing," *Int. J. Secur. Its Appl.*, vol. 13, no. 4, pp. 91–100, 2019, doi: 10.33832/ijsia.2019.13.4.09.

[6] P. Krishnadoss and P. Jacob, "OCSA: Task scheduling algorithm in cloud computing environment," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 3, pp. 271–279, 2018, doi: 10.22266/ijies2018.0630.29.

[7] H. A. Kurdi, S. M. Alismail, and M. M. Hassan, "LACE: A Locust-Inspired Scheduling Algorithm to Reduce Energy Consumption in Cloud Datacenters," *IEEE Access*, vol. 6, pp. 35435–35448, 2018, doi: 10.1109/ACCESS.2018.2839028.

[8] S. Vila, F. Guirado, J. L. Lerida, and F. Cores, "Energy-saving scheduling on IaaS HPC cloud environments based on a multi-objective genetic algorithm," *J. Supercomput.*, vol. 75, no. 3, pp. 1483–1495, 2019, doi: 10.1007/s11227-018-2668-z.

[9] P. Pirozmand, A. A. R. Hosseinabadi, M. Farrokhzad, M. Sadeghilalimi, S. Mirkamali, and A. Slowik, "Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing," *Neural Comput. Appl.*, vol. 33, no. 19, pp. 13075–13088, 2021, doi: 10.1007/s00521-021-06002-w.

[10] D. K. Shukla, D. Kumar, and D. S. Kushwaha, "WITHDRAWN: Task scheduling to reduce energy consumption and makespan of cloud computing using NSGA-II," *Mater. Today Proc.*, 2021, doi: https://doi.org/10.1016/j.matpr.2020.11.556.

[11] M. S. Sanaj and P. M. Joe Prathap, "Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere," *Eng. Sci. Technol. an Int. J.*, vol. 23, no. 4, pp. 891–902, 2020, doi: https://doi.org/10.1016/j.jestch.2019.11.002.

[12] R. Khorsand and M. Ramezanpour, "An energy-efficient task-scheduling algorithm based on a multi-criteria decision-making method in cloud computing," *Int. J. Commun. Syst.*, vol. 33, no. 9, Jun. 2020, doi: 10.1002/dac.4379.

[13] M. Hussain, L.-F. Wei, A. Lakhan, S. Wali, S. Ali, and A. Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing," *Sustain. Comput. Informatics Syst.*, vol. 30, p. 100517, 2021, doi: https://doi.org/10.1016/j.suscom.2021.100517.

[14] M. Agarwal and G. M. S. Srivastava, "Opposition-based learning inspired particle swarm optimization (OPSO) scheme for task scheduling problem in cloud computing," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 10, pp. 9855–9875, 2021, doi: 10.1007/s12652-020-02730-4.

[15] K. R. Prasanna Kumar and K. Kousalya, "Amelioration of task scheduling in cloud computing using crow search algorithm," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5901–5907, 2020, doi: 10.1007/s00521-019-04067-2.

[16] S. K. Panda and P. K. Jana, "An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems," *Cluster Comput.*, vol. 22, no. 2, pp. 509–527, 2019, doi: 10.1007/s10586-018-2858-8.

[17] D. B. L.D. and P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, 2013, doi: https://doi.org/10.1016/j.asoc.2013.01.025.

[18] T. Desai and J. Prajapati, "A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing," *Int. J. Sci. Technol. Res.*, vol. 2, no. 11, pp. 158–161, 2013, [Online]. Available: http://www.ijisi.org/final-print/nov2013/A-Survey-Of-Various-Load-Balancing-

Techniques-And-Challenges-In-Cloud-Computing.pdf
[19] S. M. A. Attallah, M. B. Fayek, S. M. Nassar, and E. E. Hemayed, “Proactive load balancing fault tolerance algorithm in cloud computing,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 10, pp. 1–11, 2021, doi: 10.1002/cpe.6172.
[20] R. A. M. Razali, R. A. B. Rahman, N. Zaini, and M. Samad, “Virtual machine migration implementation in load balancing for Cloud computing,” *2014 5th Int. Conf. Intell. Adv. Syst. Technol. Converg. Sustain. Futur. ICIAS 2014 - Proc.*, pp. 1–4, 2014, doi: 10.1109/ICIAS.2014.6869540.
[21] A. Sharma and S. K. Peddoju, “Response time based load balancing in cloud computing,” *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, pp. 1287–1293, 2014, doi: 10.1109/ICCICCT.2014.6993159.
[22] A. Composition, D. E. Interoperability, and D. Methodology, “Journal of Emerging Technologies in Web Intelligence Green Scheduling of Bag-of-tasks Applications in Cloud Data Center Using GreenCloud Simulator,” vol. 6, no. 3, 2014.
[23] M. Gopala and K. Sriram, “Challenges of Cloud Compute Load Balancing Algorithms,” *www.irjmets.com @International Res. J. Mod. Eng.*, no. 1, pp. 1186–1190, 2022, [Online]. Available: www.irjmets.com
[24] V. R. Kanakala, V. K. Reddy, and K. Karthik, “Performance Analysis of Load Balancing Techniques in Cloud Computing Environment,” 2015.
[25] M. A. Shahid, N. Islam, M. M. Alam, M. M. Su’Ud, and S. Musa, “A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach,” *IEEE Access*, vol. 8, no. c, pp. 130500–130526, 2020, doi: 10.1109/ACCESS.2020.3009184.
[26] H. E. L. Bakkali, “Load Balancing Cloud Computing : State of Art,” pp. 106–109, 2012.
[27] P. K. Tiwari, G. Rani, T. Jain, A. Mundra, and R. K. Gupta, “Load Balancing in Cloud Computing,” pp. 294–316, 2019, doi: 10.4018/978-1-7998-1021-6.ch016.
[28] T. Chaabouni and M. Khemakhem, “Energy management strategy in cloud computing: a perspective study,” *J. Supercomput.*, vol. 74, no. 12, pp. 6569–6597, 2018, doi: 10.1007/s11227-017-2154-z.
[29] N. H. Shahapure and P. Jayarekha, “Virtual machine migration based load balancing for resource management and scalability in cloud environment,” *Int. J. Inf. Technol.*, vol. 12, no. 4, pp. 1331–1342, 2020, doi: 10.1007/s41870-018-0216-y.
[30] A. P. Febina KS1, Tanuja Hemchand2, Shamimunnisabi3, Roopa P4, “A Survey on Issues and Challenges of Load Balancing Techniques in Cloud Computing Environment,” *IJRASET*, vol. 11, no. 1, 2023, doi: doi.org/10.22214/ijraset.2023.48878.
[31] N. Shah and M. Farik, “Static Load Balancing Algorithms In Cloud Computing : Challenges & Solutions,” vol. 4, no. 10, pp. 365–367, 2015.
[32] S. Saxena, K. Saxena, J. Goyal, and A. Saxena, “Cloud Computing: An Analysis on Round Robin Technique,” *SSRN Electron. J.*, pp. 244–248, 2018, doi: 10.2139/ssrn.3166544.
[33] A. Katangur, S. Akkaladevi, and S. Vivekanandhan, “Priority Weighted Round Robin Algorithm for Load Balancing in the Cloud,” in *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*, 2022, pp. 230–235. doi: 10.1109/SmartCloud55982.2022.00044.
[34] M. A. Hossain, S. Roy, S. K. Sen, and N. Hossain, “DLB3M : A noble load balancing algorithm for cloud computing services,” *J. Inf. Optim. Sci.*, vol. 42, no. 5, pp. 1087–1113, Jul. 2021, doi: 10.1080/02522667.2020.1864937.
[35] H. Chen, F. Wang, N. Helian, and G. Akanmu, “User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing,” in *2013 National Conference on Parallel Computing Technologies (PARCOMPTECH)*, 2013, pp. 1–8. doi:

10.1109/ParCompTech.2013.6621389.

[36] J. Zhou *et al.*, "Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing," *J. Cloud Comput.*, vol. 12, no. 1, 2023, doi: 10.1186/s13677-023-00453-3.

[37] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 3910–3933, 2022, doi: 10.1016/j.jksuci.2021.02.007.

[38] S. T. Milan, L. Rajabion, H. Ranjbar, and N. J. Navimipour, "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments," *Comput. Oper. Res.*, vol. 110, pp. 159–187, 2019, doi: 10.1016/j.cor.2019.05.022.

[39] T. Deepa and D. Cheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 3375–3378, 2018, doi: 10.1109/ICECDS.2017.8390086.

[40] H. Hallawi and H. He, "Complexity of combinatorial ordering genetic algorithms COFFGA and CONFGA," *AIP Conf. Proc.*, vol. 2144, no. 1, p. 50011, 2019, doi: 10.1063/1.5123127.

[41] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[42] W. Long, J. Jiao, X. Liang, and M. Tang, "Inspired grey wolf optimizer for solving large-scale function optimization problems," *Appl. Math. Model.*, vol. 60, pp. 112–126, 2018, doi: 10.1016/j.apm.2018.03.005.