International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Leveraging Swarm Intelligence Algorithms for Load Balancing in Cloud Computing Infrastructure: A Survey on Recent Advances

Stefy Methew[1], Dr. Mahaveer Kumar Sain[2], Mr.Manish Jha[3], Dr.Sonam Mittal[4]

Submitted: 02/05/2024 **Revised**: 15/06/2024 **Accepted**: 22/06/2024

Abstract: In cloud computing, load balancing is essential for maximizing system performance, guaranteeing availability, and optimizing resource use. Because of the growing complexity and scalability requirements of contemporary cloud settings, traditional static and dynamic load balancing algorithms sometimes find it difficult to adjust. Swarm Intelligence (SI) algorithms, inspired by the collective behavior of biological swarms, have emerged as effective meta-heuristic optimization techniques for task distribution and resource management. This survey explores the recent advancements in SI-based load balancing approaches, categorizing them into traditional and modern techniques. Traditional SI methods such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Grey Wolf Optimization (GWO), and BAT Algorithm offer improved efficiency but face challenges related to convergence speed and adaptability. To address these limitations, modern SI techniques like Whale Optimization Algorithm (WOA), Social Spider Algorithm (SSA), Dragonfly Optimization Algorithm (DOA), and Raven Roosting Optimization (RRO) incorporate adaptive strategies for enhanced scalability and dynamic task allocation. This research offers a thorough evaluation of different algorithms, contrasting their effectiveness, computational complexity, and practicality. The results show that cutting-edge SI techniques hold potential for flawless load balancing, quicker reaction times, and more effective utilization of cloud computing resources.

Keywords: *Cloud Computing, Load Balancing, Swarm Intelligence, Static, Dynamic, and SI-Based Approaches.*

I. Introduction

An ability of cloud computing to efficiently provide a variety of internet-based services has contributed to its rising popularity in recent years [1]. This technological advancement has changed the game for businesses when it comes to computing since it provides scalable and adaptable solutions that can keep up with the demands of today's applications [2]. Even when workloads and resources are distributed effectively in cloud systems, there may be certain difficulties. The essential load balancing technique divides incoming requests or work across many servers [3]. This guarantees constant system stability, efficient use of resources, and system improvement. Dynamic workload variations, diverse resources, changing scalability requirements, and varied application demands are just a few of the elements that make load balancing in cloud settings complex[4][5]. These problems are too big for static load-balancing techniques like least connection and round-robin to manage [6]. It can't adapt to new circumstances, don't distribute resources well, and can't handle the demands of many applications. This causes less-than-ideal resource utilisation and a drop in the system's performance[7].

In an attempt to address these issues, research and practice have focused on cloud load balancing using meta-heuristic algorithms inspired by natural processes. Algorithms like SI, GA, and a method based on the way ants find food demonstrate intelligence and adaptability by imitating real-world occurrences. This lets them distribute resources to the flow in the most equitable way possible and guarantees that workloads are distributed fairly[8]. An intriguing alternative to conventional optimization techniques is SI algorithms, which draw inspiration from the cooperative actions of natural systems like anthills, flocks of birds, and schools of fish [9]. SI algorithms may identify near-optimal solutions more accurately and efficiently than heuristics, which can get stuck in local optima. This literature review delves into the topic of SI algorithms and their application to cloud computing load balancing. It covers static, dynamic, and SI-based approaches, highlighting their effectiveness in optimizing resource distribution. The study delves into contemporary SI approaches and examines the pros and cons of these techniques. Additionally, it highlights difficulties and proposes directions for further study to enhance cloud performance and scalability.

1.Research Scholar, Faculty of Computer Science & Informatics, Maharishi Arvind University, Jaipur, Rajasthan, India, Email Id-stefymathew90@gmail.com

2.Professor, Faculty of Computer Science & Informatics, Maharishi Arvind University, Jaipur, Rajasthan, India, Email Id-mahaveersain@gmail.com.

3.Assistant Professor, Department of Computer Science, Institute of of Technology and Management, GIDA, Gorakhpur, UP India, Email Id-manishjha2k19@gmail.com. 4. Assistant Professor, Department of Computer Science,, Aishwarya College, Udaipur, Rajasthan, India, Email Id-sonammittalkota@gmail.com

A. *Motivation and Contribution*

The need for effective techniques to manage resources in order to guarantee peak performance, scalability, and energy efficiency has grown in tandem with the popularity of cloud computing. Workload distribution between virtual machines (VMs) is an important function of LB, which helps to avoid bottlenecks, reduce reaction time, and improve system dependability. Especially in ever-changing cloud settings, traditional load balancing methods have a hard time being flexible and making decisions in real time. Swarm Intelligence (SI) algorithms, inspired by collective biological behaviors, have emerged as powerful optimization techniques for solving complex scheduling and resource allocation problems. However, despite their advantages, existing SI-based methods face challenges such as slow convergence, high computational complexity, and scalability limitations. This survey is motivated by the need to explore recent advancements in SI-based load balancing techniques, analyze their effectiveness, and identify potential research directions to develop more adaptive, intelligent, and efficient cloud computing solutions. The following key focus of this paper are:

- System performance, availability, response time, and resource utilization may all be enhanced in cloud settings via load balancing.
- Explores how load balancing is essential for optimizing resource utilization, achieving high availability, and enhancing performance in cloud environments.
- Categorizes load balancing algorithms into static, dynamic, and nature-inspired approaches, highlighting their advantages and limitations.
- Explores SI algorithms inspired by biological swarms, such as ants, bees, and wolves, to optimize load balancing in cloud computing infrastructure.
- Analyzes traditional SI techniques (PSO, GA, ABC, GWO, ACO, BAT) and modern SI methods (WOA, SSA, DOA, RRO) for their efficiency in task scheduling and resource distribution.
- Identifies key challenges such as convergence speed, computational complexity, and adaptability in applying SI algorithms to cloud environments.

B. *Structure of the paper*

This paper begins with an Introduction (Section I) outlining its motivation and structure. Section II covers the Background, explaining cloud computing, the importance of load balancing, and SI's role. Section III discusses Swarm Intelligence Algorithms, including traditional and modern approaches and related challenges. A Literature Review follows in Section IV, and the paper concludes with Section V, summarizing findings and proposing future work.

II. BACKGROUND

Load balancing in cloud settings, swarm algorithms, and basic ideas and key terminologies in cloud computing are introduced in this part to establish the groundwork. We start by defining cloud computing and its fundamentals, and then we go into the critical function of load balancing in this context. We also explore the relevance and use of swarm algorithms for cloud load balancing.

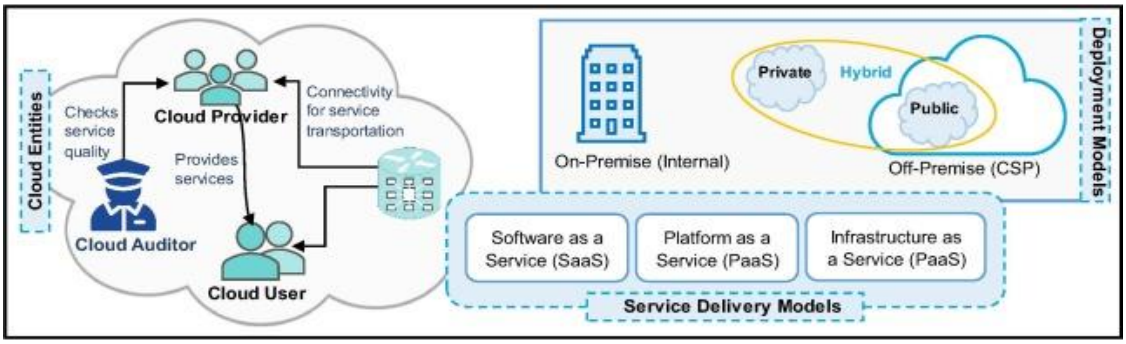

Fig. 1. Overview of cloud computing[10]

A. *Cloud Computing*

"Cloud computing" refers to a class of on-demand computer services provided by private companies like Amazon and Microsoft [11]. Computing, storage, and "SaaS" are the main aims of this concept. Cloud computing has the potential to make access to computer resources and IT services more convenient [12][13]. Cloud computing also enables the storing of vast amounts of data by several sources, like audio, video, papers, podcasts, and electronic books, among many others [14][15]. Cloud computing may be categorized into three main kinds. The models are constructed to meet the client's specifications. IaaS, PaaS, and SaaS are the paradigms. The public cloud, private cloud, hybrid cloud, and community cloud are the other four deployment methods. In Figure 1 above, cloud computing is summarized[16]. The cloud environment is managed by all cloud entities working together. To illustrate, cloud auditors play the role of cloud police by checking that CSPs provide services of a high standard and without fraud. Services in the cloud cannot be delivered to customers (cloud users) unless the provider guarantees a constant connection. With private clouds, the data center is housed within the company's network; with public clouds, it's on the internet, depending on the CSP; and with hybrid clouds, it may be situated in either the public or private cloud.

Table I provides a summary of both types of models with different aspects.

TABLE I. SERVICE VS DEPLOYMENT MODELS OF CLOUD COMPUTING WITH DIFFERENT ASPECTS

Aspect	**Cloud Services models**			**Cloud deployment models**			
	Software as a Service (SaaS)	***Platform as a Service (PaaS)***	***Infrastructure as a Service (IaaS)***	***Public Cloud***	***Private Cloud***	***Hybrid Cloud***	***Community Cloud***
Definition	Software applications delivered over the internet.	Platform for developers to build and deploy apps.	Virtualized computing resources over the internet.	Cloud resources shared publicly.	Dedicated cloud resources for a single organization.	Combination of public and private cloud infrastructure.	Cloud resources shared among organizations with common goals.
Primary Use	End-user applications like email, CRM, etc.	Application development, testing, and deployment.	Hosting and managing IT infrastructure.	Cost-effective and scalable public usage.	Security and control for sensitive data.	Flexibility and workload optimization.	Collaboration across specific communities or sectors.
Examples	Google Workspace, Salesforce, Dropbox.	AWS Elastic Beanstalk, Microsoft Azure App Services.	AWS EC2, Microsoft Azure VM, Google Compute Engine.	AmazonWeb Services,Microsoft Azure.	On-premises VMware, OpenStack.	AWS Outposts, Azure Arc.	Government or research organizations.
Ownership	Managed by service providers.	Managed by service providers for developers.	Managed by providers; users control resources.	Service provider owns and manages.	Owned and operated by the organization.	Shared ownership (provider + organization).	Shared by multiple organizations.
Cost	Subscription-based, predictable.	Pay-per-use or subscription-based.	Pay-as-you-go for infrastructure.	Low cost due to shared resources.	Higher cost for dedicated infrastructure.	Balanced cost depending on workloads.	Shared cost among participating entities.
Scalability	Fully scalable with no user intervention.	Highly scalable for app development.	Scalable as per user configuration.	Highly scalable, but shared environment.	Limited scalability based on resources.	Scalable across both public and private clouds.	Scalable depending on community needs.
Security	Managed by the provider, less control for users.	Provider-managed; developers focus on apps.	Full user control over infrastructure security.	Moderate, based on shared infrastructure.	High, due to isolated environment.	Balanced security (shared/private environments).	High, with shared governance policies.

Customization	Limited; predefined software features.	High; supports custom app development.	Full control over infrastructure setup.	Limited to provider's services.	Fully customizable.	Customizable to some extent.	Moderate; depends on community agreements.

B. Role of Load Balancing in Cloud Computing

A system's load balancing mechanism distributes the burden across its several resources. As a result, each resource in a cloud-based system has to be performing about the same amount of work at any one time for the load distribution to be optimal. Fundamentally, various methods for balancing requests are required after provide the application solution more quickly [17]. All cloud providers rely on automated load balancing services, which let customers scale up their resources by adding more CPUs or RAM. These services are customizable to meet the specific requirements of the client's company. Load balancing, then, works to improve performance and, secondly, the availability of resources in a cloud[18].

1) Load-Balancing Model

A load balancer is shown as an essential part of the load-balancing paradigm in Figure 2. In order to distribute user requests fairly across available virtual machines, this load balancer employs load-balancing techniques. The load balancer is vital in distributing requests to the best virtual machine. Concurrently, the duty of managing tasks, including their submission for load balancing, is taken up by a data center controller. The load balancer then uses load-balancing algorithms to distribute work to VMs based on their ability to complete the tasks at hand. Also, the VMs themselves are supervised by the VM manager. Virtualization, which seeks to enable the sharing of valuable hardware resources among VMs, is one of the most significant components of cloud computing. VMs are software representations of actual computers that make it easier to run software and running operating systems. An international user base makes queries at random, and these VMs patiently handle all of them. Most importantly, these requests must be assigned to VMs for processing. Critical issues emerge when some VMs experience an undue load while others are unoccupied or have little workload. As a result of the uneven allocation of workloads, QoS might be diminished. Dissatisfaction, brought on by a decline in QoS, may lead customers to uninstall the system entirely.

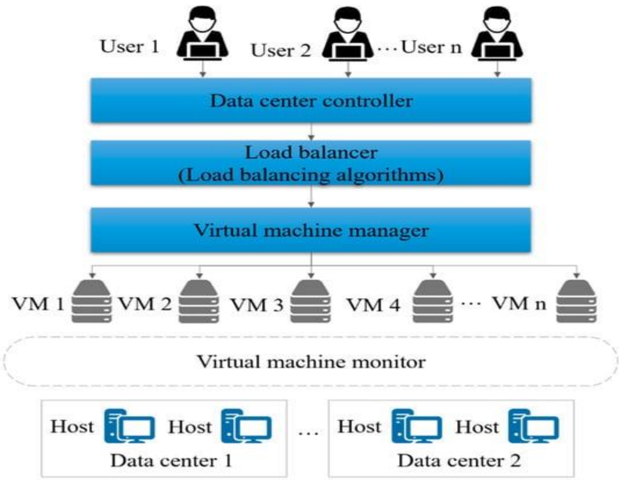

Fig. 2. Load-balancing model[19].

Creating and maintaining VMs heavily relies on the function of a hypervisor or VM Monitor (VMM). Storage, provisioning, migration, and multiplexing are among the crucial functions that the VMM makes possible. All of these functions are essential to efficient load balancing. Pourghebleh and Hayyolalam [20] highlight the significance of load balancing in handling two critical operations: allocation of resources and scheduling of tasks. Energy conservation, decreased resource utilization costs, higher resource utilization, high resource availability, preservation of cloud computing's flexibility, and decreased carbon emissions are all outcomes of these processes converging. Evaluation and direction of load-balancing algorithms may be achieved via the use of several measures. These metrics provide useful information on the efficiency, effectiveness, and operation of load balancing in a computer system. Here is a summary of these metrics in Table II [19]:

TABLE II. PERFORMANCE METRICS OF LB IN CC

Metrics	Summarizing
Throughput	Throughput measures the pace at which a system completes operations over a period of time. This metric is utilized to assess the processing capability and efficiency of a system.
Makespan	The makespan value is correlated with the longest completion time or the allocation of user resources. It indicates how long it will take to do all of the tasks and offers information on system efficiency.
Response time	A task's whole execution time, including processing, communication, and queueing delays, is captured by response time. The speed at which tasks are completed is a reflection of the user experience.

Scalability	A system's scalability may be defined as its capacity to maintain a constant workload distribution across all of its nodes regardless of the size of the network. It makes no difference how many nodes are in the network; a scalable algorithm can handle load distribution efficiently.
Migration time	The time required to move tasks from overburdened hosts to underutilized ones is known as the migration time. The responsiveness of the system and the utilization of resources are affected.
Fault tolerance	The ability of an algorithm to maintain load balancing in the event of connection or node failure is measured by its fault tolerance. It guarantees system performance and stability under trying situations.
Imbalance degree	Disparity in workload allocation across virtual machines or nodes is quantified by the degree of imbalance. Optimal system performance is enhanced by distributing workloads evenly.
Energy consumption	The network's energy consumption indicates how much energy it uses overall. By minimizing resource use and avoiding overheating, load balancing helps save energy.

2) Activities involved in load balancing

Task scheduling and VM allocation according to needs is the workload in cloud computing. The following steps comprise load balancing [3]:

- **Identification of user task requirements:** An amount of resources required to perform user tasks scheduled on a VM is determined at this step.
- **Identification of resource details of a VM:** A VM's resource data is validated in this way. It shows the current resource use of the virtual machine and the unallocated resources. with this stage, VM may be categorized as either balanced, overloaded, or underloaded with respect to a threshold.
- **Task scheduling:** Once the resource characteristics of a VM have been settled, a scheduling technique assigns tasks to appropriate resources on the most suitable VMs.
- **Resource allocation:** Tasks with a due date are assigned the necessary resources. To do this, a strategy for allocating resources is being used. The scheduling algorithm's and allocation policy's effectiveness dictates the load balancing algorithm's strength.
- **Migration:** The cloud load balancing procedure isn't complete without migration, which is a crucial step in the process. The entities that are examined distinguish between VM migration and task migration, the two kinds of cloud migration. There are two forms of virtual machine migration: live VM migration and non-live migration. When a VM becomes overloaded, it is possible to fix the problem by migrating it to a different physical host.

3) Load balancing algorithms in CC

Several popular load balancing strategies are used to improve CC's performance. The three main groups into which these algorithms are often subdivided based on their foundational context are static algorithms, dynamic algorithms, and algorithms influenced by nature.

- **Static Load Balancing (SLB) Algorithms:** The way the load balancing algorithms perform in a static environment depends on a system state's properties and functions as well as prior knowledge[21]. In runtime, static-based techniques do not account for changes in a load. Examples of SLB algorithms like Round Robin Algorithm, Weighted Round Robin Algorithm, Random Allocation Algorithm, Least Connections Algorithm, and Hash-Based Allocation Algorithm etc.
- **Dynamic Load Balancing (DLB) Algorithms:** Load balancing is an area where these algorithms excel and are quite versatile. When applied to a dynamic setting, load balancing algorithms remember the system's condition in the past, in contrast to SLB algorithms[22]. Examples of DLB algorithms like Centralized Dynamic Scheduling, Distributed, Honeybee Foraging Algorithm, Biased Random Sampling, Adaptive Load Balancing Algorithm, Least Response Time Algorithm and Weighted Dynamic Balancing Algorithm etc.
- **Nature-inspired Load Balancing (NLB) Algorithms:** The genetic mechanism by which bees discover honey is one example of a biological process or activity that such algorithms mimic[23]. Load balancing in CC is achieved by mathematically modeling these processes to mimic their natural counterparts. Examples of NLB algorithms like: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC), Firefly Algorithm, Bat Algorithm, Cuckoo Search Algorithm, Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA) etc.

C. Swarm Intelligence (SI) Support Load Balancing in Cloud Computing

Cloud resources could be wasted, or service performance might be impaired due to problems with underperforming or overperforming services caused by inadequate scheduling. The concept of integrating meta-heuristic methods into task scheduling was successfully motivated by the need to allocate few resources to complicated and ever-changing incoming workloads, or cloudlets [9]. Applications for routing and algorithms connected to specialized job scheduling make use of SI. The most talked-about problem with cloud computing, "load balancing," is really the result of these two SI applications. By drawing inspiration from insects, SI makes load balancing easy[24]. SI is based on the idea that social insects work collaboratively to solve complex problems. It appropriately suggested a really clever and decentralized method, which is just what cloud computing needs after effectively handling a load. As a result, these autonomous, sociable, and self-aware bug behaviors have advanced into a model for handling the difficult load balancing problem in cloud environments [25].

III. Swarm Intelligence Algorithms

Insect and mammal social behavior models provide the basis for SI algorithms, which explain a number of interrelated problem-solving techniques. These programs act like biological swarms. Multiple swarms collaborate in a search space using SI algorithms to discover a solution. In particular, the SI is a meta-heuristic that has been refined by modeling problem-solving after the actions of actual swarms of insects. When optimizing complicated problems with a dispersed structure, SI techniques come in handy. Also, their use in systems with elastic and flexible characteristics won't affect the structure as a whole. This led to the widespread use of SI methods for better resource scheduling in the cloud[26].

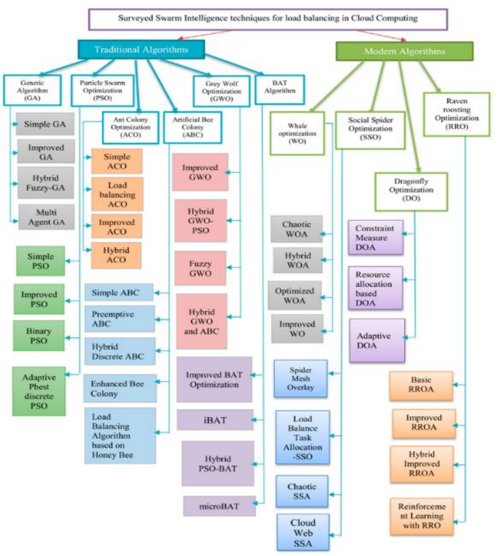

Fig. 3. Swarm intelligence algorithms in LB[9]

Figure 3 shows that there are 2 main types of SI load balancing algorithms included in this survey: conventional algorithms and modern algorithms. Here are the specifics of their variants.

A. Traditional SI Algorithms for Load Balancing

A field of swarm intelligence is driven by the desire to analyze and implement "collective intelligence" systems. This intelligence is atmospherically dispersed, coordinated, and decentralized. When it comes to cloud computing, the load balancing issue may be easily solved using swarm-based methods [27]. Figure 3 displays a number of chosen Swarm Intelligence-based load balancing algorithms:

1) Genetic Algorithm (GA)

GAs are fundamental AI methods that model computer programs after biological processes. Then, these digital algorithms learn much like biological systems [28]. As a result, swarm intelligence is used in conjunction with genetic algorithms to develop optimization approaches. Their main focus is on creating an environment that mimics the natural process of evolution [29]. The use of genetic algorithms is a reflection of biological level expansion. Here, in nature, genetics transmit the passed-down instructions for an organism. Chromosomes include genetic coding that contains this biological information. In instance, these codes may be used to identify an organism's traits. Darwin came to the conclusion that an organism's chances of survival are entirely dependent on the traits it inherits from its parents. To reproduce and pass on their genes, organisms that are "most fit" to survive in a given environment will be selected. The GA [30] involves three important parameters: crossover, mutation, and selection.

2) Particle Swarm Optimization (PSO)

An initial idea of PSO was derived from the swarm of birds in flight. Determining which location to land at is a hard challenge, but the flock of birds eventually finds one. Several things determine the landing. The presence or absence of carnivores is one of these determinants, along with the accessibility of food. Consequently, the birds land simultaneously after flying in synchrony until they discover the optimal landing spot. Several additional approaches' performance metrics inform PSO's integration of workload with reduced reaction time for each individual task. PSO is simpler to understand and implement, and it has less computational expenses [31]. It resolves the complex issue of cloud load balancing by operating in adaptive mode, which allows it to manage the convergence vs. divergence tradeoff and find the optimal solution.

3) Ant Colony Optimization (ACO)

ACO is a metaheuristic algorithm that estimates possible solutions to complicated problems efficiently; it is population-centered. Studying ant colonies served as an inspiration for this method. It finds workable solutions to a problem by coordinating the efforts of a hive of software agents, which are like artificial ants. We find the best route by making a weighted graph. The software agents, like ants moving in a straight line, discover paths across the graph and construct solutions progressively. This approach of creating solutions presumptively allows agents to alter the values of a set of parameters linked to graph components (nodes or arcs) while the program is running. The next section presents a few of the algorithms that are based on ant colonies. Additional algorithms like IMaxMin-ACO, CUDA-based ACO, IACO, and modified ACO are also suggested to interested readers [32].

4) Artificial Bee Colony (ABC)

The three bee groups that make up the colony—working bees, scouts, and observers—serve as the model for ABC. Each food source is expected to have one fake bee. After returning from foraging for food, the bees will dance in this area, leading us to believe that the number of working bees in the colony is directly proportional to the amount of food sources in the vicinity of the hive. After this bee has its food supply cut off, it acts as a scout to find other sources of sustenance. Various research studies also show alternative variants of the ABC algorithm, including ABC with EA, IABC, GB guided ABC, modified ABC, hybrid ABC, memetic ABC, and crossed and mutant ABC [33].

5) Grey Wolf Optimization (GWO)

The GWO algorithm was created for the purpose of job allocation in an effort to achieve efficient load balancing. This program mimics how wolves stalk their prey. This algorithm's four-tiered load balancing strategy is reminiscent of a pack of multi-level wolves. Alpha, beta, delta, and omega are the names given to these four levels. The leader of the group, regardless of gender, is at the top level when it comes to decision-making. Supporters on level 2 assist the leader in making decisions and upholding authority within the pack. Scouts are supposed to monitor the restrictions at Level 3 for safety. The pack's substitutes are included in Level 4; it consume last. Other variations of the GWO algorithm, including DGWO, TLBO-GW, Modified GWO, hybridized versions GWO, PGWO [34][35], and others, are also provided in other research work in addition to algorithms covered in the following section.

6) BAT Algorithm

Echolocation behavior served as inspiration for BAT, an optimization technique. Bats find their prey by making a variety of noises. Bats are able to attract a wide variety of prey in frequency form through their sonar capabilities. The distance is then calculated using this frequency once all the signals have been gathered [34]. The concept may be tweaked to provide load balancing by focusing the node's efforts on making coordination more localized, as it functions autonomously. The advantages and disadvantages of the conventional SI algorithms that were reviewed are shown in Table III.

TABLE III. PROS AND CONS OF TRADITIONAL SI ALGORITHMS.

Algorithm	**Pros**	**Cons**
GA	It provides an improved load balancing solution while also making better use of resources.	Complexity in computation is increased by the many computation steps. Efficiency decreases as the search space is enlarged. Priority time is not provided to the same extent.
PSO	Its utilization rate is higher. A physical machine receives the load instead of an overloaded virtual machine.	The issue determines the algorithm's performance.
ACO	Distribution of workloads among nodes that maximizes utilization of available resources.	It takes more time for convergence to occur.
ABC	Raises the boundary for maximum throughput.	The solution increases the computing cost and the process slows down when used in a sequential manner due to the lack of supporting material.
GWO	Efficient use of resources and a disparity in manpower between global and local search activities facilitate and accelerate convergence.	The grey wolves' roles are of equal importance, which goes against their established social order.
BAT	Precision and efficiency are two of its strongest points. When compared to other options, it has lower processing expenses.	Fast convergence happens because there is no preliminary mathematical study that connects the parameters to convergence rates.

B. Modern SI Algorithm for Load Balancing

Traditional SI algorithms have been adequately developed to address the challenges of load balancing in cloud-based job allocation, as we have seen in the previous sections [35]. The usefulness of even the most effective of these algorithms is constrained by a number of problems. As a result, cutting-edge methods are crucial because of the time it takes for convergence to occur, the difficulty of implementing it, and the difficulty of guaranteeing scalability. This section discusses roosting optimization methods and covers the most competitive cloud-based swarm intelligence load-balancing strategies, such as the raven, spider, whale, and dragonfly.

1) Whale Optimization Algorithm (WOA)

As a novel SI optimization technique, the Whale Optimization technique was proposed by Australian researchers Mirjalili and Lewis in 2016 [36]. The model accurately depicts the humpback whale population's natural hunting behavior, which includes behaviors such as circling, loop updating location, and random hunting techniques. The following are the primary procedures of WOA[37]:

- **Initialization phase:** The creation of the population is done at random during this phase.
- **Fitness calculation phase:** This step involves calculating a fitness function. This is how the fitness function is computed. Based on the test results, the best whale (agent) is chosen.
- **Encircling prey phase:** According to popular belief, this is the mechanism that fixes the prey's location. With the present strategy being the most secure, humpback whales encircle their prey. In response to the current best agent, other whales adjust their places accordingly.

2) Social Spider Algorithm (SSA)

Scientists from all across the world have identified almost 50,000 species of spiders. Three groupings have been identified by researchers. Social, sub social, and colony spiders are among these varieties. Among these three categories, social spiders are distinct. These spiders gather in clusters and exchange messages with each other. Yu et al.[38] presented SSA as a prominent methodology for optimization strategies, drawing inspiration from the foraging habits of social spiders [39].

3) Dragonfly Optimization Algorithm (DOA)

DA's behavior is reminiscent of a swarming dragonfly [40]. The two primary causes of their swarming are foraging and relocation (specifically, static swarm and dynamic swarm). Dragonflies travel in tiny groups across a limited region in pursuit of other species. This type of swarming is characterized by abrupt changes and social motions. In contrast, dynamic swarming is characterized by a significant number of dragonflies aggregating into a singular community and continuing to migrate in a predetermined direction over an extended period. In order to guarantee the convergence of dragonfly individuals, dragonflies should modify their weights during the optimization phase to accommodate the transition from intensifying to diversifying[41].

4) Raven Roosting Optimization Algorithm (RROA)

Typically, ravens will return to their roosts just before dusk and then leave in closely spaced flocks at dawn the next day [42]. Ravens like these settle on a roosting spot early on and don't budge from there. The next step is to randomly assign each raven a starting position to find food. In the end, this leads to calculating the ravens' fitness levels. At the conclusion of the evaluation, the one with the best solution is named the leader. Following this determination, a predetermined quantity of ravens is chosen. Along with the leader, these chosen ravens go from the nest to locate the ideal spot, which may be quite a distance away. The accompanying ravens begin by finding the best possible answer by measuring the hemisphere's radius. It take any arbitrary point after evaluating. The benefits and drawbacks of the contemporary SI algorithms that were described are shown in Table IV. Table V summarizes the discussed SI-based load balancing algorithms, including their main purpose, the field of application, and targeted issue(s), along with an appropriate reference.

TABLE IV. PROS AND CONS OF MODERN ALGORITHMS.

Algorithm	Pros	Cons
WOA	A rate of successfully performed jobs is increased.	Convergence is postponed since it often fails in the first iterative cycle.
SSA	Enhances a number of QoS metrics by using the global best match.	Lucrative job execution rates decline as the number of similar VMs grows.
DOA	Offers a highly efficient solution that quickly converges to the global optimal one.	Response rates are shown to be mediocre overall, rather than quick and quick in the absence of a nearby answer.
RROA	Ensures that neither overloading nor underloading occurs.	There is a decrease in the rate of task completion in beginning iterations.

TABLE V. SUMMARY OF THE SI BASED LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING.

Authors	Algorithm	Main Objective	Area of Application	Targeted Issue(s)
Hodzic and Mrdovic, (2023) [30]	Genetic algorithm	Improve response times in cloud settings by using load balancing.	Gives serious thought to the need to acquire imagery in the actual world.	The goal is to use permutation encoding to expand the possible range of request IDs. The reaction time is improved.
Datiri and Li, 2023 [43]	Particle Swarm Optimization Algorithm	Optimize the load.	Achieves minimal task execution and transfer time by making use of target functionalities.	Efficiency optimization for power consumption and operation.
He, (2022) [44]	Ant colony Optimization Algorithm	Determine the best way to schedule tasks in order to optimize the load.	Complete the scheduling task as effectively as possible while maximizing the system's load management needs.	Performance is improved.
Gupta and Bhadauria, (2023) [45]	LBA_Honey Bee	Keeping capacity in a cloud-based setting.	Seek to prevent both under- and over-utilization.	The makespan, execution time, response time, and load standard deviation are all decreased.
Mohammadian et al., (2023) [46]	Artificial Bee Colony	VM Load Balancing in the Cloud.	The efficiency of global searches and convergence rates are confirmed.	Quick convergence with great adaptability.
Kruekaew and Kimpan, (2022) [47]	Hybrid artificial bee colony algorithm with multi-objective	The cloud allows for flexible scheduling of tasks.	Accelerates convergence while enhancing performance.	Facilitates better exploitation.
Sefati, Mousavinasab and Zareh Farkhady, (2022) [48]	GWO	Keeping capacity in a cloud-based setting.	Suggest load balance and resource distribution.	Decreases the makespan.
Ouhame, Hadi and Arifullah, (2020) [49]	Hybrid GWO_ABC	Load balancing is being used by the system that allocates resources.	The VM's cloud computing resource allocation mechanisms are now 1.25 percent more accurate and reliable.	Reduces waste, boosts efficiency, and shortens the average time it takes for a network to run.
Ullah, Nawi and Khan, (2020) [34]	BAT Algorithm	To enhance performance in a cloud-based setting.	Select VMoptimally.	Satisfy QoS.
Saoud and Recioui, (2022) [50]	Hybrid WOA-BAT	Balancing loads in a cloud setting.	Fast convergence.	Exchanges between basic WO's exploitation and exploration features.
Arul Xavier and Annadurai, (2019) [51]	SS Cloud Web Algorithm	Load balancing non-pre-emptive tasks.	Improve QoS requirements.	Assign tasks and resources to the population while imposing quality limits.

Neelima and Reddy (2020) [52]	DOA	Management of resources to ensure workload parity.	Task scheduling, load balancing, and resource allocation all saw significant speed improvements.	Datacenter problems with task scheduling.
Torabi and Safi-Esfahani, (2018) [53]	Improved RRO	The distribution of work should be enhanced.	Time spent waiting, average response, and performance may all be improved.	The goal is to stop the problem of early convergence.

C. *Issues Related to Swarm Intelligence based Load balancing Algorithms*

This section provides an overview of the primary concerns with the Cloud Load Balancing Algorithm that is based on SI and needs to be resolved prior to its implementation in the cloud. The algorithm's performance might suffer if these problems aren't fixed. Here are the main points of these difficulties:

- Load balancing strategies based on static SI are what this implies for static clouds. These algorithms work well in secure cloud settings. Nevertheless, load balancing algorithms based on SI provide a formidable challenge due to the inherent uncertainty of cloud systems.
- Load balancing algorithms that rely on SI tend to be centralized. The existence of a single point of failure is a serious concern with large-scale cloud infrastructures. Distributed SI-based complex load balancing algorithms are notoriously difficult to create.
- Particle swarms, honeybees, and artificial ants are just a few examples of the many agents produced by load balancing algorithms based on SI. When it comes to load balancing, these agents are a big assistance. Moreover, these bots keep an eye on the cloud. Setting these agents to work in tandem is no easy feat.
- A huge number of agents are generated via load balancing algorithms that are based on swarm intelligence. Agents like these keep an eye on the nodes in the cloud from all across the network. The cloud network's performance may degrade due to the increased overhead caused by such a huge number of agents.

IV. Literature Review

Cloud computing has been the subject of numerous research. Load balancing, resource scheduling, resource allocation, service broker restrictions, and other cloud computing-related issues are covered in these surveys. The polls about cloud computing load balancing are detailed in the section that follows.

This study, Sharma et al. (2024) proposes an improved initialization approach for the Tuni-cate Swarm Optimization (TSO) algorithm, leveraging strategic heuristics. Specifically, Shortest Job First Priority (SJFP) and Earliest Completion Time (ECT) heuristics are employed to initialize TSO. Both of the proposed SJFP-TSO and ECT-TSO algorithms are evaluated on the performance metrices including minimizing makespan, total execution time, load balancing, and energy consumption. Comparative analysis shows the superiority of the proposed algorithms over conventional TSO method[54].

This study, Yadav et al. (2024) suggests a Multi-objective Optimization technique to overcome these issues. GA, PSO and ACO Algorithm are utilized in the study to schedule tasks with multiple objectives. Makespan, a critical parameter measuring an overall time necessary to accomplish all jobs, is the primary focus of the study since it assesses the performance of various algorithms in great detail. In our research, PSO produces a makespan of 1.1988, GA produces 1.3025, while ACO remarkably achieves a minimum makespan of 0.8736 after 200 iterations. The comparison’s findings demonstrate that ACO is better to PSO and GA, making it the ideal choice for cloud task scheduling algorithm optimization. The significant difference in makespan values demonstrates ACO’s superior capacity to explore the solution space, converge effectively, and offer plans that minimize the overall amount of time required to accomplish the task. The findings of this study are crucial because they demonstrate the advantages of ACO for scheduling several objectives efficiently at once[55].

This study, Singhal et al. (2024) introduces a load balancing algorithm based on Rock Hyrax that uses QoS parameters to solve problems with power efficiency and local maxima. The approach decreases makespan 10%-15 % and total energy usage in data centers by 8%-13% when compared to conventional scheduling algorithms. The results demonstrate that the Rock Hyrax-based load balancing algorithm is successfully enhancing data center performance and energy efficiency. Additionally, it showcases the algorithm's capacity to enhance system performance by optimizing resource allocation[56].

This research, Rajpoot, Singh and Pant (2023) aims to study and compare an efficiency of various nature-inspired load balancing algorithms, like ACO, PSO, GA, and ABC. The paper analyzes these algorithms' features, benefits, limitations, and challenges and evaluates their effectiveness in load balancing in distributed systems[57].

Prasanna Kumar et al. (2023) introduces a fresh method for balancing loads known as FFBSO. It combines BSO's optimization skills with the FF's search space reduction capabilities. Tasks are represented by birds, and VMs are destination food patches in BSO, which draws influence from the group behavior of birds. Tasks on the cloud are considered to be non-preemptive and autonomous. Still, the BSO algorithm finds the optimal places for VMs and assigns jobs to them. At 35s makespan, 99% maximum resource utilization, and 13ms average reaction time, the FFBSO algorithm outperformed the alternatives in the simulations [58].

Al Reshan et al. (2023) emphasis of this research is on GWO and PSO. In order to reap the benefits of both fast convergence and global optimization, this research proposes a GWO-PSO combination strategy. Combining these two methods improves system efficiency and resource allocation, which in turn solves the load-balancing issue. By reducing overall reaction time and achieving globally optimized rapid convergence, the study's findings show promise when contrasted with more traditional approaches. In comparison to other approaches, the proposed one decreases reaction time by 12% on average. In addition, the target function of the suggested GWO-PSO algorithm yields the best optimal value, which enhances PSO's convergence to 97.253% [59].

This research, Shu and Gao (2023) presents a DMLBP for data centre networks (DCNs) with the goal of efficiently handling elephant flow conflicts in DCNs. The experimental outcomes demonstrate that a DMLBP algorithm improves network throughput and bandwidth utilization while decreasing network propagation time in comparison to the existing ECMP and SA methods[60].

Alghamdi (2022) The goal of this study was to develop ANN-BPSO, a cost-effective binary version of the popular PSO method, to optimize the distribution of computing resources in the cloud and decrease latency. Compared to the conventional BPSO task scheduling method, our technique improves performance by 22% according to resource utilization and 33% according to mean time[61].

Table VI provides a comparison of various load balancing across different cloud environments and platforms using optimization algorithms.

TABLE VI. SUMMARY OF THE RELATED WORK FOR LOAD BALANCING IN CLOUD COMPUTING USING OPTIMIZATION ALGORITHMS

References	Methodology	Objective	Performance	Advantage	Challenges	Future Work
Sharma et al. (2024)	Improved initialization for TSO using SJFP and ECT heuristics	Minimize makespan, total execution time, load balancing, and energy consumption	SJFP-TSO and ECT-TSO outperform conventional TSO in all metrics, ensuring better resource utilization.	Improved load balancing and energy efficiency.	Limited analysis on scalability for larger data centers.	Explore hybrid approaches for further enhancing initialization techniques.
Yadav et al. (2024)	Multi-objective optimization using GA, PSO, and ACO algorithms	Optimize makespan in cloud task scheduling.	ACO achieves the best makespan (0.8736), outperforming PSO (1.1988) and GA (1.3025).	ACO demonstrates superior convergence and exploration capability.	High computational cost for ACO in larger, dynamic cloud environments.	Extend the study to include real-time scheduling scenarios.
Singhal et al. (2024)	Rock Hyrax-based load balancing algorithm	Improve QoS, reduce energy consumption and makespan.	Reduces makespan by 10%–15% and energy consumption by 8%–13%.	Effectively balances workloads while saving energy.	May require fine-tuning for specific QoS constraints.	Test on hybrid cloud environments and extend for fault-tolerant systems.
Rajpoot et al. (2023)	Comparative analysis of nature-inspired algorithms:	Evaluate effectiveness of nature-inspired algorithms in	Detailed comparison of algorithm	Provides insights into the strengths	No experimental validation or dataset-	Incorporate emerging algorithms for next-generation

	ACO, PSO, GA, and ABC	load balancing.	capabilities and challenges.	and limitations of algorithms.	specific analysis.	distributed systems.
Prasanna Kumar et al. (2023)	Hybrid FFBSO combining Firefly and Bird Swarm Optimization algorithms	Minimize response time, maximize resource utilization, reduce makespan.	Achieved 13ms average response time, 99% resource utilization, and 35s makespan.	Combines strengths of FF and BSO, achieving high optimization efficiency.	Scalability and adaptability to dynamic workloads not thoroughly evaluated.	Enhance scalability and integrate real-time task migration mechanisms.
Al Reshan et al. (2023)	Hybrid GWO-PSO for load balancing in cloud computing.	Achieve fast convergence, optimize resource allocation, reduce response time.	Reduced response time by 12% compared to traditional methods; improved convergence to 97.253%.	Enhanced convergence rate and response time optimization.	Limited exploration of dynamic task arrivals and load variability.	Test the hybrid model on large-scale heterogeneous cloud systems.
Shu and Gao (2023)	Dynamic multipath load balancing algorithm based on PSO (DMLBP) in DCN	Cope with elephant flow conflicts in DCN and improve bandwidth utilization.	Increased bandwidth utilization, network throughput, and reduced propagation delay compared to ECMP and SA algorithms.	Addresses elephant flow conflicts effectively in data centers.	Requires optimization for fault tolerance and packet loss handling.	Investigate real-time adaptive strategies in large-scale DCNs.
Alghamdi et al. (2022)	ANN-BPSO (Artificial Neural Networks with Binary PSO).	Minimize task scheduling time and balance cloud computing resources.	Increased resource utilization by 22% and decreased mean scheduling time by 33%.	Cost-effective with significant improvements in resource usage.	ANN-based model might face scalability and adaptability issues for diverse tasks.	Develop multi-objective optimization with dynamic task handling.

V. CONCLUSION AND FUTURE WORK

Cloud load balancing using SI algorithms has shown to be very effective due to its dynamic adaptability, better resource utilization, and higher scalability. Traditional SI approaches such as GA, PSO, ACO, ABC, GWO, and BAT Algorithm have demonstrated efficiency in task distribution but suffer from issues like slow convergence and high computational overhead. In contrast, modern SI techniques like WOA, SSA, DOA, and RRO incorporate adaptive mechanisms that improve real-time decision-making and optimize workload allocation. Despite their advantages, challenges such as handling high-dimensional data, ensuring fault tolerance, and reducing execution latency remain. Future research should focus on hybrid SI models that integrate deep learning, reinforcement learning, and blockchain technology to enhance decision-making, security, and predictive analytics in cloud load balancing. Additionally, exploring quantum-inspired SI algorithms and energy-aware scheduling techniques can further improve performance and sustainability in large-scale cloud infrastructures.

REFERENCES

[1] B. Pourghebleh, A. Aghaei Anvigh, A. R. Ramtin, and B. Mohammadi, “The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments,” *Cluster Comput.*, 2021, doi: 10.1007/s10586-021-03294-4.

[2] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, “Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends,” *Concurr. Comput. Pract. Exp.*, 2022, doi: 10.1002/cpe.6698.

[3] S. Afzal and G. Kavitha, “Load balancing in cloud computing – A hierarchical taxonomical classification,” *Journal of Cloud Computing*. 2019. doi: 10.1186/s13677-019-0146-7.

[4] A. Pradhan, S. K. Bisoy, and A. Das, “A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment,” *Journal of King Saud University - Computer and Information Sciences*. 2021. doi: 10.1016/j.jksuci.2021.01.003.

[5] R. Arora, S. Kumar, N. Jain, and M. T. Nafis, “Revolutionizing Healthcare with Cloud Computing: Superior Patient Care and Enhanced Service Efficiency,” *SSRN*, 2022, doi: http://dx.doi.org/10.2139/ssrn.4957197.

[6] A. and P. Khare, “Cloud Security Challenges : Implementing Best Practices for Secure SaaS Application Development,” *Int. J. Curr. Eng. Technol.*, vol. 11, no. 6, pp. 669–676, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.11.

[7] Y. Lohumi, D. Gangodkar, P. Srivastava, M. Z. Khan, A. Alahmadi, and A. H. Alahmadi, “Load Balancing in Cloud Environment: A State-of-the-Art Review,” *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3337146.

[8] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, “A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3065308.

[9] M. A. Elmagzoub, D. Syed, A. Shaikh, N. Islam, A. Alghamdi, and S. Rizwan, “A survey of swarm intelligence based load balancing techniques in cloud computing environment,” *Electronics (Switzerland)*. 2021. doi: 10.3390/electronics10212718.

[10] 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, Jul. 2022, doi: 10.1016/j.jksuci.2021.02.007.

[11] S. Murri, S. Chinta, S. Jain, and T. Adimulam, “Advancing Cloud Data Architectures: A Deep Dive into Scalability, Security, and Intelligent Data Management for Next-Generation Applications,” *Well Test. J.*, vol. 33, no. 2, pp. 619–644, 2024, [Online]. Available: https://welltestingjournal.com/index.php/WT/article/view/128

[12] A. Goyal, “Optimising Cloud-Based CI/CD Pipelines: Techniques for Rapid Software Deployment,” *TIJER – Int. Res. J.*, vol. 11, no. 11, pp. a896–a904, 2024.

[13] Vashudhar Sai Thokala, “Scalable Cloud Deployment and Automation for E-Commerce Platforms Using AWS, Heroku, and Ruby on Rails,” *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 349–362, Oct. 2023, doi: 10.48175/IJARSCT-13555A.

[14] R. Arora, S. Gera, and M. Saxena, “Impact of Cloud Computing Services and Application in Healthcare Sector and to provide improved quality patient care,” *IEEE Int. Conf. Cloud Comput. Emerg. Mark. (CCEM), NJ, USA, 2021*, pp. 45–47, 2021.

[15] B. Boddu, “CLOUD DBA STRATEGIES FOR SQL AND NOSQL DATA MANAGEMENT FOR BUSINESS-CRITICAL APPLICATIONS,” *https://ijcem.in/wp-content/uploads/CLOUD-DBA-STRATEGIES-FOR-SQL-AND-NOSQL-DATA-MANAGEMENT-FOR-BUSINESS-CRITICAL-APPLICATIONS.pdf*, vol. 7, no. 1, p. 8, 2022.

[16] 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*. 2022. doi: 10.1016/j.jksuci.2021.02.007.

[17] T. K. K. and S. Rongala, “Implementing AI-Driven Secure Cloud Data Pipelines in Azure with Databricks,” *Nanotechnol. Perceptions*, vol. 20, no. 15, pp. 3063–3075, 2024, doi: https://doi.org/10.62441/nano-ntp.vi.4439.

[18] K. Mishra and S. K. Majhi, “A state-of-art on cloud load balancing algorithms,” *Int. J. Comput. Digit. Syst.*, 2020, doi: 10.12785/IJCDS/090206.

[19] P. Li, H. Wang, G. Tian, and Z. Fan, “Towards Sustainable Cloud Computing: Load Balancing with Nature-Inspired Meta-Heuristic Algorithms,” *Electronics*, vol. 13, no. 13, p. 2578, Jun. 2024, doi: 10.3390/electronics13132578.

[20] B. Pourghebleh and V. Hayyolalam, “A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things,” *Cluster Comput.*, 2020, doi: 10.1007/s10586-019-02950-0.

[21] J. P. Yang, “Efficient load balancing using active replica management,” in *2016 International Conference on Applied System Innovation, IEEE ICASI 2016*, 2016. doi: 10.1109/ICASI.2016.7539944.

[22] T. M. Tawfeeg *et al.*, “Cloud Dynamic Load Balancing and Reactive Fault Tolerance Techniques: A Systematic Literature Review (SLR),” *IEEE Access*. 2022. doi: 10.1109/ACCESS.2022.3188645.

[23] V. Arulkumar and N. Bhalaji, “Performance analysis of nature inspired load balancing algorithm in cloud environment,” *Journal of Ambient Intelligence and Humanized Computing*. 2021. doi: 10.1007/s12652-019-01655-x.

[24] C. Udatha and G. Lakshmeeswari, “An Adaptive Load Balancing using Particle Swarm Optimization for Cloud Task Scheduling,” *Int. J. Eng. Trends Technol.*, 2023, doi: 10.14445/22315381/IJETT-V71I9P204.

[25] U. Singhal and S. Jain, “An Analysis of Swarm Intelligence based Load Balancing Algorithms in a Cloud Computing Environment,” *Int. J. Hybrid Inf. Technol.*, 2015, doi: 10.14257/ijhit.2015.8.1.22.

[26] B. A. S. Emambocus, M. B. Jasser, and A. Amphawan, “A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms,” *IEEE Access*, 2023, doi: 10.1109/ACCESS.2022.3233596.

[27] S. Aslam and M. A. Shah, “Load balancing algorithms in cloud computing: A survey of modern techniques,” in *2015 National Software Engineering Conference, NSEC 2015*, 2016. doi: 10.1109/NSEC.2015.7396341.

[28] X. Zhou *et al.*, “Load balancing prediction method of cloud storage based on analytic hierarchy process and hybrid hierarchical genetic algorithm,” *Springerplus*, 2016, doi: 10.1186/s40064-016-3619-x.

[29] M. S. Kabir, K. M. Kabir, and R. Islam, “Process of Load Balancing In Cloud Computing Using Genetic Algorithm,” *Electr. Comput. Eng. An Int. J.*, 2015, doi: 10.14810/ecij.2015.4206.

[30] L. Hodzic and S. Mrdovic, “Using Genetic Algorithms for Load Balancing in Cloud Computing,” in *2023 29th International Conference on Information, Communication and Automation Technologies, ICAT 2023 - Proceedings*, 2023. doi: 10.1109/ICAT57854.2023.10171261.

[31] S. E. Dashti and A. M. Rahmani, “Dynamic VMs placement for energy efficiency by PSO in cloud computing,” *J. Exp. Theor. Artif. Intell.*, 2016, doi: 10.1080/0952813X.2015.1020519.

[32] R. Gao and J. Wu, “Dynamic load balancing strategy for cloud computing with ant colony optimization,” *Futur. Internet*, 2015, doi: 10.3390/fi7040465.

[33] A. Kumar, D. Kumar, and S. K. Jarial, “A review on artificial bee colony algorithms and their applications to data clustering,” *Cybern. Inf. Technol.*, 2017, doi: 10.1515/cait-2017-0027.

[34] A. Ullah, N. M. Nawi, and M. H. Khan, “BAT algorithm used for load balancing purpose in cloud computing: an overview,” *Int. J. High Perform. Comput. Netw.*, 2020, doi: 10.1504/ijhpcn.2020.110258.

[35] K. Bhargavi, B. Sathish Babu, and J. Pitt, “Performance Modeling of Load Balancing Techniques in Cloud: Some of the Recent Competitive Swarm Artificial Intelligence-based,” *J. Intell. Syst.*, 2020, doi: 10.1515/jisys-2019-0084.

[36] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[37] F. S. Gharehchopogh and H. Gholizadeh, “A comprehensive survey: Whale Optimization Algorithm and its applications,” *Swarm Evol. Comput.*, 2019, doi: 10.1016/j.swevo.2019.03.004.

[38] J. J. Q. Yu and V. O. K. Li, “A social spider algorithm for global optimization,” *Appl. Soft Comput. J.*, 2015, doi: 10.1016/j.asoc.2015.02.014.

[39] J. Qiu, J. Xie, F. Cheng, X. Zhang, and L. Zhang, “A hybrid social spider optimization algorithm with differential evolution for global optimization,” *J. Univers. Comput. Sci.*, 2017.

[40] C. M. Rahman and T. A. Rashid, “Dragonfly algorithm and its applications in applied science survey,” *Computational Intelligence and Neuroscience*. 2019. doi: 10.1155/2019/9293617.

[41] Z. Amini, M. Maeen, and M. R. Jahangir, “Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing,” *Int. J. Networked Distrib. Comput.*, 2018, doi: 10.2991/ijndc.2018.6.1.4.

[42] A. Brabazon, W. Cui, and M. O’Neill, “The raven roosting optimisation algorithm,” *Soft Comput.*, 2016, doi: 10.1007/s00500-014-1520-5.

[43] D. D. Datiri and M. Li, “Effects of Particle Swarm Optimisation on a Hybrid Load Balancing Approach for Resource Optimisation in Internet of Things,” *Sensors*, 2023, doi: 10.3390/s23042329.

[44] J. He, “Cloud Computing Load Balancing Mechanism Taking into Account Load Balancing Ant Colony Optimization Algorithm,” *Comput. Intell. Neurosci.*, 2022, doi: 10.1155/2022/3120883.

[45] A. Gupta and H. S. Bhadauria, “Honey Bee Based Improvised BAT Algorithm for Cloud Task Scheduling,” *Int. J. Comput. Networks Appl.*, 2023, doi: 10.22247/ijcna/2023/223310.

[46] V. Mohammadian, N. J. Navimipour, M.

Hosseinzadeh, and A. Darwesh, “LBAA: A novel load balancing mechanism in cloud environments using ant colony optimization and artificial bee colony algorithms,” *Int. J. Commun. Syst.*, 2023, doi: 10.1002/dac.5481.

[47] B. Kruekaew and W. Kimpan, “Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm with Reinforcement Learning,” *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3149955.

[48] S. S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, “Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation,” *J. Supercomput.*, 2022, doi: 10.1007/s11227-021-03810-8.

[49] S. Ouhame, Y. Hadi, and Arifullah, “A Hybrid Grey Wolf Optimizer and Artificial Bee Colony Algorithm Used for Improvement in Resource Allocation System for Cloud Technology,” *Int. J. online Biomed. Eng.*, 2020, doi: 10.3991/ijoe.v16i14.16623.

[50] A. Saoud and A. Recioui, “Hybrid algorithm for cloud-fog system based load balancing in smart grids,” *Bull. Electr. Eng. Informatics*, 2022, doi: 10.11591/eei.v11i1.3450.

[51] V. M. Arul Xavier and S. Annadurai, “Hybrid starling social spider algorithm for energy and load aware task scheduling in cloud computing,” *Int. J. Innov. Technol. Explor. Eng.*, 2019, doi: 10.35940/ijitee.i8596.078919.

[52] P. Neelima and A. R. M. Reddy, “An efficient load balancing system using adaptive dragonfly algorithm in cloud computing,” *Cluster Comput.*, 2020, doi: 10.1007/s10586-020-03054-w.

[53] S. Torabi and F. Safi-Esfahani, “A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing,” *J. Supercomput.*, 2018, doi: 10.1007/s11227-018-2291-z.

[54] I. Sharma, R. Gupta, and P. Singh, “Improved Initialization for Tunicate Swarm Optimization in Cloud Task Scheduling,” in *2024 IEEE Region 10 Symposium (TENSYMP)*, IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/TENSYMP61132.2024.10752302.

[55] D. Yadav and B. Abraham Amal Raj, “An efficient swarm intelligence algorithm for Multi-Objective Task Scheduling Optimization in the Context of Cloud Computing,” in *2024 International Conference on Automation and Computation (AUTOCOM)*, IEEE, Mar. 2024, pp. 148–152. doi: 10.1109/AUTOCOM60220.2024.10486073.

[56] S. Singhal *et al.*, “Energy Efficient Load Balancing Algorithm for Cloud Computing Using Rock Hyrax Optimization,” *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3380159.

[57] N. K. Rajpoot, P. Singh, and B. Pant, “Nature-Inspired Load Balancing Algorithms for Resource Allocation in Cloud Computing,” in *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2023*, 2023. doi: 10.1109/CISES58720.2023.10183630.

[58] K. R. Prasanna Kumar, S. Gm, N. Yamsani, T. M. Kiran Kumar, and A. K. Pani, “A Novel Energy-Efficient Hybrid Optimization Algorithm for Load Balancing in Cloud Computing,” in *IEEE 1st International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics, AIKIIE 2023*, 2023. doi: 10.1109/AIKIIE60097.2023.10390196.

[59] M. S. Al Reshan *et al.*, “A Fast Converging and Globally Optimized Approach for Load Balancing in Cloud Computing,” *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3241279.

[60] Y. Shu and D. Gao, “A Dynamic Multipath Load Balancing Algorithm Based on Particle Swarm Optimization in DCN,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2023. doi: 10.1109/ICSESS58500.2023.10293042.

[61] M. I. Alghamdi, “Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO),” *Sustain.*, 2022, doi: 10.3390/su141911982.