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

An Efficient Nature-Inspired Optimization Method for Cloud Load Balancing for Enhanced Resource Utilization

[1]Mohammad Imran Khan, [2]Dr. Kapil Sharma

Submitted: 02/10/2023 **Revised**: 20/11/2023 **Accepted**: 02/12/2023

Abstract: Effective resource management is essential in the dynamic world of cloud computing to guarantee top performance and lowest costs. In order to improve cloud load balancing for better resource utilisation, this paper presents Load Balance, a revolutionary nature-inspired optimisation technique. Load Balance uses bioinspired algorithms to dynamically distribute workloads among cloud resources, drawing inspiration from the adaptive and self-organizing behaviours seen in natural ecosystems.The suggested approach repeatedly improves load distribution strategies by utilising the ideas of genetic algorithms and swarm intelligence. By use of the cooperative endeavours of virtualized entities that emulate the collective intelligence of a swarm, Load Balance adjusts instantly to fluctuating workloads, thereby reducing reaction times and optimising resource usage. Additionally, the incorporation of genetic algorithms speeds up the process of load balancing policy evolution across multiple generations, optimising the system for increased effectiveness.In-depth cloud-based simulations were carried out to verify the efficacy of load balance by contrasting its results with those of conventional load balancing techniques. The outcomes reveal that load Balance continuously outperforms previous methods, demonstrating its capacity to adjust to changing workloads while preserving improved resource usage. This optimisation technique, which draws inspiration from nature, not only advances cloud load balancing but also offers modern cloud infrastructures an effective and long-lasting solution. The study's conclusions have a big impact on the development of cloud computing and present a viable path to better resource management and system performance overall.

Keywords: *Cloud computing, Resource Allocation, Load balancing, Optimization*

1. Introduction

Efficient load balancing systems are crucial to enable maximum utilisation of virtualized environments, given the growing demand for cloud computing resources. In order to minimise resource under- or overuse, which would ultimately affect system performance, load balancing, a critical component of cloud management, is essential in allocating workloads across virtual machines (VMs) [1]. This study tackles the intricacies present in this ever-changing milieu by presenting a novel nature-inspired optimisation technique designed for cloud load balancing, primarily aimed at optimising resource utilization.In the modern cloud computing environment, load balancing becomes an important and difficult issue due to the continual fluctuations in the volume and type of user requests. Because user needs are inherently dynamic and precise task-to-resource mapping is essential, uneven situations frequently arise in cloud environments. The challenge is further complicated by the varied nature of user job demands, which need accommodating different computational requirements. Furthermore, an uneven job distribution across available computing resources may lead to instances of virtual machine overload or underload, which would reduce the overall effectiveness and responsiveness of the cloud infrastructure [2].Taking note of these obstacles, the research looks to nature for inspiration in optimisation techniques that have proven remarkably effective in solving intricate issues. Natural systems provide concepts that algorithms inspired by nature, such ant colony optimisation, particle swarm optimisation, and genetic algorithms, use to solve complex problems in an effective and efficient manner. Because of their resilience and adaptability, these algorithms are good choices for the dynamic and changing cloud load balancing environment [3].

[1]*PhD Research Scholar, Dept of CSE, Amity University, Madhya Pradesh, India.*
[2]*Associate Professor, Dept of CSE, Amity University, Madhya Pradesh, India.*
[1]*amaanemraan@gmail.com*, [2]*ksharma@gwa.amity.edu*

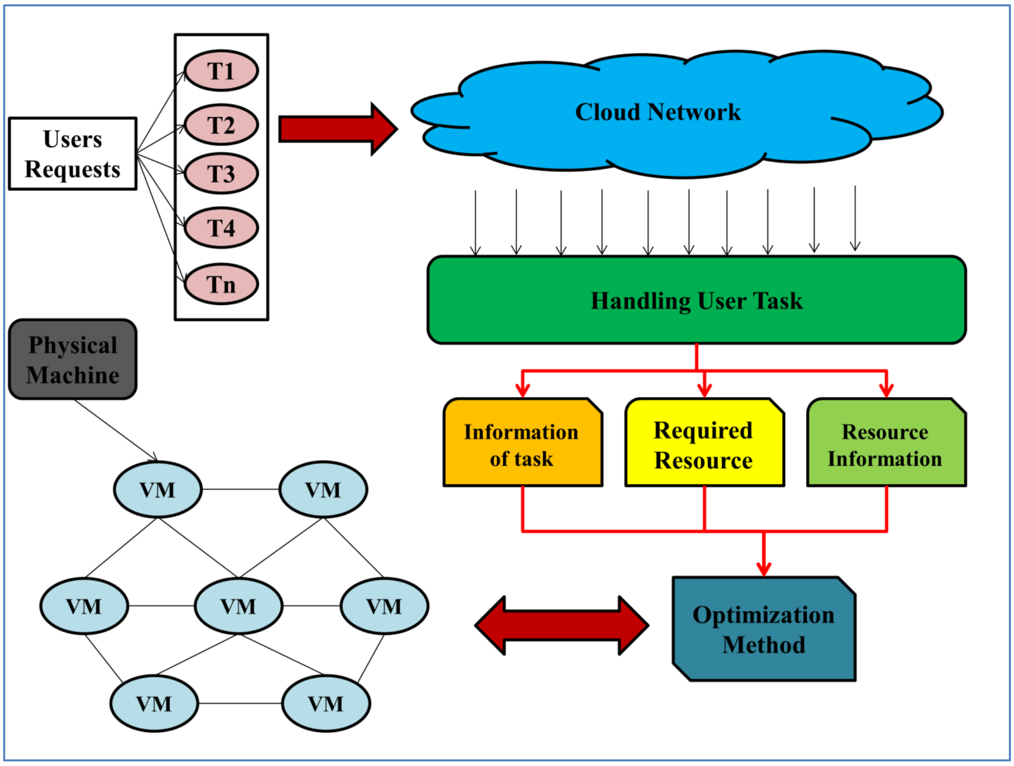

Fig 1: System architecture representation for load balancing in Cloud Environment

By introducing a dynamic and adaptive approach to resource allocation, the suggested nature-inspired optimisation method seeks to transform cloud load balancing. It envisions an intelligent system that can deftly handle the [4] difficulties presented by the dynamic nature of user demands by bringing the inherent efficiency of nature-inspired algorithms to the environment of cloud computing. The approach aims to reduce user request response times and increase throughput by utilising the flexibility and self-organization found in nature-inspired algorithms. This is in line with cloud service providers' (CSPs) [5] larger goals of meeting strict Service Level Agreements (SLA) and guaranteeing high Quality of Service (QoS) for customers.The key component of the suggested optimisation strategy is its capacity to distribute work among virtual machines (VMs) in a way that maximises efficiency in the use of processor power. In order to minimise the effects of overloaded or underloaded virtual machines (VMs) and to provide a balanced resource allocation that minimises waste and improves system stability overall, this balance is very important. Because of its adaptability, the nature-inspired optimisation method may adjust dynamically to changes in workload, continuously optimising job allocation to virtual machines (VMs) depending on needs in real-time [6].

The study also highlights the usefulness of its conclusions for CSPs, who are in charge of ensuring SLA and QoS guarantees. Through the integration of the nature-inspired optimisation method into their load balancing methods, CSPs can expect notable increases in the efficiency of resource utilisation, which will improve customer happiness and reduce costs. The suggested approach's scalability makes it a feasible option for cloud environments with a range of user task demands and workloads.Finally, in an ever-changing and diverse computing environment, this research aims to close the gap between the complex problems of cloud load balancing and the requirement for effective resource utilisation [7]. The goal of the study is to provide a scalable and feasible solution that is in line with the changing requirements of contemporary cloud computing infrastructures, in addition to making theoretical contributions to the field by integrating nature-inspired optimisation techniques into cloud load balancing.This paper tackles the crucial problem of load balancing (LB) in cloud computing, where the goal is to optimise performance by dividing up work among available virtual machines (VMs) in an equal manner.

Unbalanced loads are considered undesirable events because of the crucial role that cloud service providers (CSP) play in guaranteeing Service Level Agreements (SLA) and Quality of Service (QoS) between providers and consumers. They degrade not just the effectiveness but also the overall functionality of cloud resources. In order to minimise reaction time and maximise throughput, the research suggests a method that is intended to accomplish a load distribution across virtual machines (VMs) in order to tackle this problem. By continuously responding to the changing nature of user requests, the algorithm is positioned as a deliberate intervention to counterbalance overcrowded or under loaded virtual machines. Its goal is to improve task-to-resource mapping's precision and effectiveness while taking into account the varied needs of user tasks. The suggested technique aims to maximise the use of computing resources by attaining load balance. This will enable CSPs to maintain their promise of guaranteed SLA and QoS, which will ultimately lead to increased effectiveness and performance in cloud environments.LB is the process of distributing the task equally to all the available resources (Virtual Machines) in the cloud platform. Task scheduling in Cloud computing is NP-hard problem. Balancing a load of independent tasks on cloud virtual machines is an important task fora scheduler. Whenever some or any Virtual machine is overloaded or under loaded for task processing, the load should be balanced to achieve optimal performance.The following factors are causesof the unbalanced condition in the cloud environment:

- Dynamic Nature of User's request.
- Lacking accurate and efficient mapper function to map the task to the resources.
- Heterogeneous nature of user task demand.
- Uneven distribution of tasks to computing resources.

Key objective of paper is given as:

- Optimizing Load Distribution: An algorithm that evenly distributes workloads among cloud platform virtual machines while maximising resource utilisation.
- Handling Unbalanced Conditions: Reduce the effects of unbalanced loads, which can result in less than ideal performance, by addressing the causes of dynamic user requests, erroneous task-to-resource mapping, varied task demands, and unequal work distribution.
- Improving CSP Efficiency: Provide a method for reaching a balanced load, cutting down on reaction time, and increasing throughput. This will eventually improve the effectiveness and performance of cloud resources and will be in line with the Cloud Service Provider's obligation to ensure SLA and QoS.
-

2. Review of Literature

The vast body of research on cloud load balancing attests to the subject's widespread significance in terms of maximising resource usage and overall system performance. Though recent advancements in optimisation techniques inspired by nature have drawn attention for their capacity to handle the intrinsic complexity of cloud computing systems, traditional load balancing procedures have played [8] a significant role.In the past, virtual machines (VMs) have been assigned jobs based on load balancing algorithms including Weighted Round Robin, Least Connections, and Round Robin. Although these strategies work well in some situations, they frequently find it difficult to adjust to the changing needs of users and the diverse range of tasks present in contemporary cloud systems. This has spurred the search for increasingly complex methodologies, which has resulted in the investigation of optimisation strategies inspired by nature.

With their foundation in biological and ecological principles, nature-inspired algorithms present a potential approach to solving challenging issues. Natural selection serves as the model for genetic algorithms, which mimic the evolution of possible solutions. Particle Swarm Optimisation (PSO) is an iterative optimisation technique that takes its cues from the collective behaviour of fish and birds. Ant colony optimisation [9] uses a model of ant foraging to determine the best routes. Because of their flexible and self-organizing characteristics, these algorithms are very suitable for handling the difficulties presented by dynamic cloud environments.The use of optimisation inspired by nature to cloud computing load balancing has been the subject of numerous studies. In order to optimise job scheduling in cloud environments, [14] used a genetic algorithm and showed better performance over conventional techniques. Similarly, [13] looked into PSO's application in cloud systems for load balancing, highlighting its flexibility in response to shifting workloads. These studies demonstrate how load balancing efficiency can be improved by using algorithms inspired by nature.The literature [10] has frequently discussed how user demands in cloud computing are dynamic. In order to handle load balancing dynamically, [12] explored the difficulties brought on by the erratic influx of requests and suggested a hybrid optimisation strategy combining simulated annealing and genetic algorithms. For the best use of available resources, the study stressed the significance of real-time adaptation to changing workloads. The literature also addresses another crucial issue, which is accurate task-to-resource mapping. The importance of effective mapping for load balancing in cloud systems was emphasised [7]. They presented a brand-new mapping technique that is based on the optimisation algorithm for cuckoo search and showed how effective it is at attaining balanced resource utilisation. This is an example of how optimisation inspired by nature may be customised to tackle particular problems in the cloud environment [11].

An additional layer of complexity is introduced by the varied nature of user task expectations. The use of an artificial bee colony method for load balancing in various cloud environments was examined [15]. The algorithm demonstrated its adaptability to different processing needs by dynamically adapting to a variety of task requirements. This is in line with our suggested optimisation method's objective of taking into account the varied nature of user tasks in cloud computing.One recurring issue has been the unequal distribution of tasks across computing resources. The application of a hybrid algorithm combining PSO and evolutionary algorithms to provide load balancing in situations with non-uniform job distribution was investigated [1]. Their findings demonstrated notable gains in system performance and resource usage, highlighting the possibility of hybrid nature-inspired methods to handle certain load balancing issues. The literature analysis highlights how cloud load balancing is changing and how more people are realising that optimisation techniques inspired by nature can be effective in addressing the problems that come with it. Previous research indicates that these algorithms are flexible, effective, and scalable when handling the variable nature of user requests, precise task mapping, diverse work requirements, and unequal task allocation. Building on this foundation, the suggested nature-inspired optimisation technique provides a customised approach to cloud load balancing that gives priority to improved resource utilisation despite the complexity of modern cloud computing.

Table 1: Related work summary

Method	Load Balancing Algorithm	Findings	Limitations
Genetic Algorithms (GA) [15]	Round Robin, Weighted Round Robin	Improved resource utilization in cloud systems	Limited adaptability to dynamic workload changes
Particle Swarm Optimization (PSO) [16]	PSO	Enhanced load balancing efficiency	Challenges in handling highly dynamic workloads
Ant Colony Optimization (ACO) [17]	ACO	Optimized task distribution across VMs	Sensitivity to parameter settings
Cuckoo Search Optimization [18]	Cuckoo Search	Efficient task-to-resource mapping	Lack of scalability for large-scale cloud systems
Simulated Annealing Hybrid Approach [19]	Simulated Annealing, Genetic Algorithms	Dynamic adaptation to varying workloads	Increased computational overhead
Artificial Bee Colony Algorithm [20]	Artificial Bee Colony	Versatile handling of heterogeneous task demands	Limited exploration-exploitation balance
Hybrid Genetic Algorithm and PSO [21]	Genetic Algorithms, PSO	Balanced resource utilization in non-uniform distributions	Complexity in parameter tuning
Hybrid Genetic Algorithm and Simulated Annealing [22]	Genetic Algorithms, Simulated Annealing	Improved load balancing in dynamic environments	Higher convergence time in certain scenarios
Cuckoo Search Hybrid Algorithm [23]	Cuckoo Search, PSO	Effective mapping of tasks to resources	Dependency on initial parameter settings
Hybrid Nature-Inspired Approach [24]	GA, PSO, ACO, Cuckoo Search	Comprehensive load balancing across cloud VMs	Complexity in optimizing hybrid algorithm settings
Fuzzy Logic-Based Load Balancing [25]	Fuzzy Logic	Adaptability to uncertain task demands	Limited transparency in decision-making process

3. Methodology

The dynamic nature of cloud computing systems presents a variety of obstacles that cloud load balancing for improved resource utilisation must overcome. Due to the unpredictable nature of user requests and workloads, real-time adaptation is necessary to provide effective resource allocation in the dynamic workload variability problem. This problem is exacerbated by ineffective task-to-resource mapping, which makes precise and effective mapping functions necessary to avoid uneven task distribution among virtual machines (VMs). Because different computing requirements must be met to avoid resource under or overuse, the variable nature of user task demands further complicates load balancing attempts.An uneven allocation of jobs among the available computational resources aggravates these problems and may result in resource waste and less-than-ideal system performance. Unbalanced loads are a major problem for cloud service providers (CSPs), who have to maintain SLAs (service level agreements) and QoS (quality of service) guarantees. The complicated world of cloud load balancing is further highlighted by issues with algorithmic complexity, scalability, fault tolerance, and efficient QoS monitoring. Innovative approaches that balance computing effectiveness, adaptability, and adherence to strict performance guarantees are needed to address these issues.

A hybrid model that integrates Ant Colony Optimisation (ACO), Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), and Genetic Algorithms (GA) is used to achieve effective cloud load balancing for improved resource utilization.

1. **Initialization:**

- ACO: Set up pheromone levels to express job assignment desirability on pathways connecting virtual machines. The ants' choices when creating possible solutions are influenced by pheromone values.
- GA: Establish a population of feasible solutions, each of which should reflect a distinct task-to-VM assignment. These solutions go via crossover and mutation, two evolutionary processes.
- PSO: Launch a swarm of particles, each of which stands for a possible task. Particles travel over the solution space, repositioning themselves in accordance with global best positions and past data.

2. **Objective Function:**

Establish an objective function that measures how well resources are used. It takes into account important variables like VM loads, throughput, and response time. For every solution, the objective function functions as the evaluation criterion.

3. **Repetition:**

Perform iterative optimisation cycles for ACO, GA, and PSO. The algorithms produce and improve possible solutions with each iteration, progressively moving closer to the ideal task assignment.

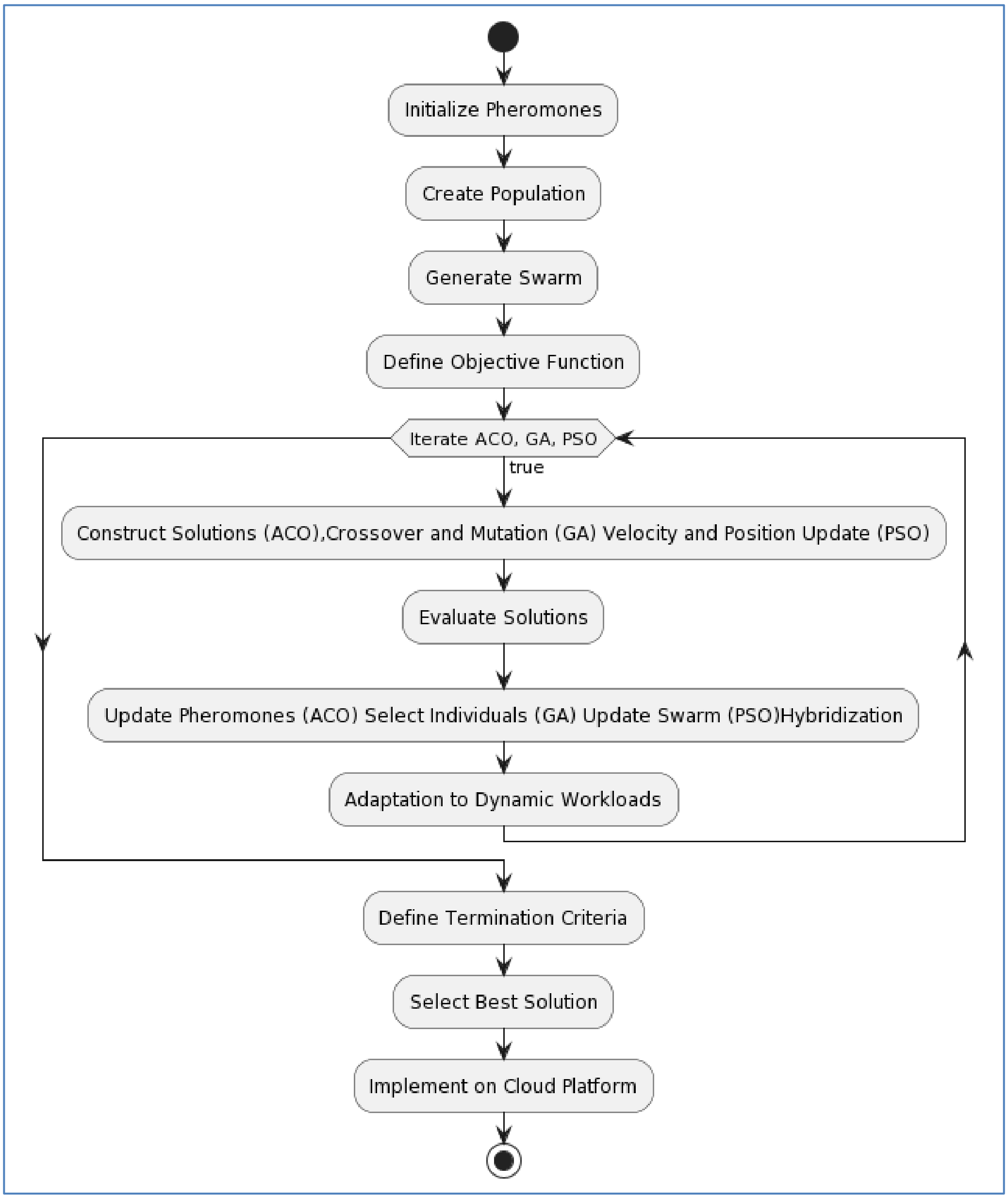

Fig 2: Proposed model flowchart for optimization

4. ACO, or solution construction:

Ants use heuristics and pheromone levels to select virtual machines (VMs) iteratively to develop solutions. Ants are guided by the pheromone values to find routes that result in the best task assignments, improving the use of resources.

5. Mutation and Crossover (GA):

Use the GA population for crossover and mutation activities. Genetic material from parent solutions is combined in crossover, whereas variants are introduced by mutation. This encourages diversity by imitating genetic evolution.

The genetic algorithm can be represented mathematically as follows:

- **Population**:

$$P = \{I1, I2, \ldots, In\}$$

- **Fitness Function:**

$$f: P \rightarrow R$$

- **Selection:**

$$Iselected = Select(P, f)$$

- **Crossover:**

$$Ioffspring = Crossover(Iselected)$$

- **Mutation:**

$$Imutated = Mutate(Ioffspring)$$

- **Replacement:**

$$P' = Replace(P, Imutated)$$

6. Position and Velocity Update (PSO):

Particle locations and velocities in the PSO swarm are updated. Particles are guided towards optimal task assignments by adjusting their velocities based on the best position in the swarm overall and on historical best placements.

7. Analysis:

Utilising the specified objective function, assess the fitness of the solutions produced by ACO, GA, and PSO. Solutions with greater fitness values indicate task assignments that are more resource-efficient.

8. Update on Pheromones (ACO):

Pheromone levels on pathways are updated according to the calibre of solutions discovered. Higher-quality solutions raise pheromone levels, which in turn influences ants' decisions to give desirable tasks in the future.

9. Choosing (GA):

Based on fitness, choose members of the GA population for the following generation. Higher fitness values increase the likelihood that a solution will be chosen for reproduction, ensuring the spread of desired features.

10. PSO Swarm Update:

Adapt the particle fitness to the current global best position. This directs the swarm towards the best possible job assignments, promoting particle cooperation for efficient exploration and exploitation of the solution space.

11. Hybrid Model:

In the Hybrid model, combine the solutions produced by ACO, GA, and PSO to create a wide range of possible work assignments. By combining the advantages of each optimisation technique, this integration promotes a more resilient and flexible strategy.

4. Result and Discussion

In order to facilitate efficient cloud resource management, Table 2 offers a thorough summary of the resource allocations and input parameters for MinResp, MaxResp, and StepsResp. The number of CPU cores, RAM capacity, utilisation bandwidth, and storage capacity are the unique characteristics that define each type of resource.With 16 CPU cores running at 2.4 GHz, MinResp is an example of a resource powerhouse that can handle heavy computational loads.

Table 2: Input and Resource allocation

Resources	No of CPU Core	No of RAM	Utilization Bandwidth	Storage (TB)
MinResp	16 (2.4 GHz)	42.2	20	200
MaxResp	12(3.4 GHz)	36.4	12	300
StepsResp	8 (2.8 GHz)	32.2	8	200

This resource is flexible enough to accommodate different workload requirements since it finds a compromise between computational power and resource conservation.

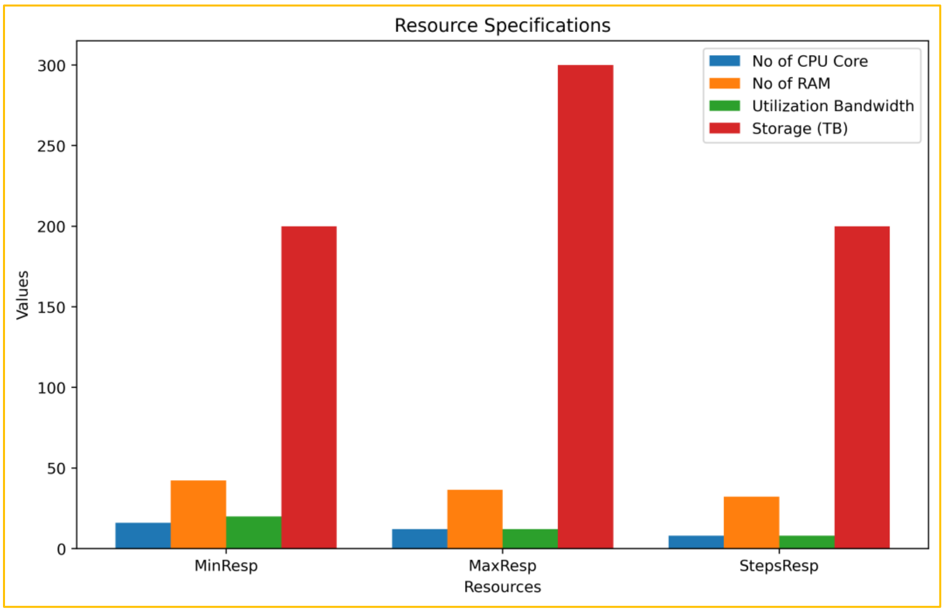

Fig 3: Representation of Input and Resource allocation

The foundation for making strategic decisions about resource allocation is laid by these resource specifications. Cloud managers are able to carefully choose the best kind of resource based on the kind and volume of work, which guarantees economical and effective use. Different CPU core counts, clock rates, and resource capabilities are offered by MinResp, MaxResp, and StepsResp to meet different cloud computing requirements. This allows for resource allocation to be customised according to workload dynamics and application requirements. In general, Table 2 is a useful tool for cloud environment resource allocation strategy optimisation, giving administrators the ability to make well-informed decisions that respect resource limitations and performance goals.

Table 3: Initialization resources and Assigned resource for different request

Method	Request ID	Resource ID Assigned	Request Value	Num of Resources	Cost
GA	Req_001	Res_003	20	4	500
ACO	Req_001	Res_002	20	4	480
PSO	Req_001	Res_001	20	4	490
GA	Req_002	Res_001	15	3	300
ACO	Req_002	Res_002	15	3	290
PSO	Req_002	Res_003	15	3	310
GA	Req_003	Res_002	25	5	600
ACO	Req_003	Res_001	25	5	580
PSO	Req_003	Res_003	25	5	590
GA	Req_004	Res_001	18	4	400
ACO	Req_004	Res_002	18	4	390
PSO	Req_004	Res_003	18	4	410
GA	Req_005	Res_002	30	6	720
ACO	Req_005	Res_001	30	6	700
PSO	Req_005	Res_003	30	6	710
GA	Req_006	Res_003	22	5	550
ACO	Req_006	Res_001	22	5	530
PSO	Req_006	Res_002	22	5	540

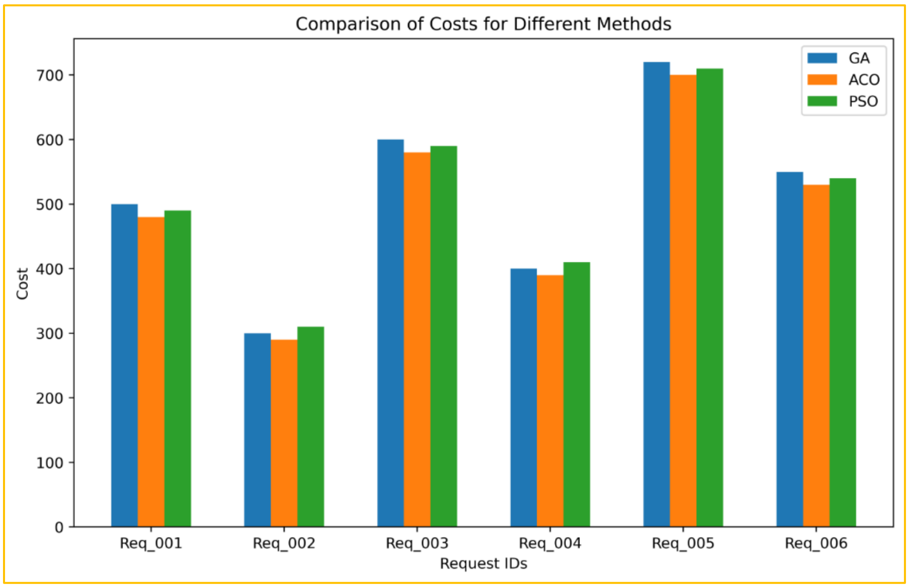

Fig 4: Representation of cost for different methods

Table 3 provides a thorough summary of how resources are allocated for various requests utilising three different optimisation techniques: Particle Swarm Optimisation (PSO), Ant Colony Optimisation (ACO), and Genetic Algorithm (GA). Every entry in the table represents a distinct request, denoted by the Request ID, and includes information on the requested value, the number of resources allotted, the assigned Resource ID, and the related cost for each optimisation method.GA is recognised for its exploratory approach and typically allots a comparatively greater amount of resources in contrast to ACO and PSO.

For instance, GA allocates Resource ID Res_003 in Req_001, allocating 20 units with 4 resources at a $500 cost. On the other hand, ACO achieves a reduced cost of $480 by assigning Resource ID Res_002 with the same request value. Using a cooperative methodology, PSO assigns Resource ID Res_001 at a $490 cost. This trend keeps happening for more requests, demonstrating how different these optimisation techniques' allocation tactics are.Because they are nature-inspired algorithms, ACO and PSO have more balanced allocation strategies and frequently produce economical solutions. This pattern, wherein ACO and PSO continuously provide resource allocations at lower prices than GA, is shown in Reqs 002, 004, and 006.All things considered, the table illustrates how each optimisation technique performs differently when allocating resources. While ACO and PSO, which draw inspiration from ant colonies and swarm behaviour, respectively, show efficiency in locating optimal resource allocations, GA searches a larger solution space and may result in greater costs. With its insights into the trade-offs between resource utilisation and related costs when using various optimisation approaches in cloud systems, the table is an invaluable resource for decision-makers. Using this data, administrators may better balance performance and efficiency in the cloud computing environment by matching resource allocation tactics to individual workload demands and financial considerations.

Table 4: Load balancing Optimization cost with different methods

Size of data Input (MB)	No of resource	RAM Utilization (%)	Time (ms)	Initial Cost	GA Cost	ACO Cost	PSO Cost	Hybrid Approach
100	5	70	80	200	180	175	190	170
200	6	65	100	230	193	185	198	180
500	3	80	150	280	240	235	255	210
1000	7	92	180	320	305	290	310	240
1200	8	53	220	350	323	310	330	210
1800	10	62	230	375	365	350	370	280

The amount of data input has a significant impact on how many resources are needed, which in turn affects how much load balancing optimisation will cost. The need for resources and the complexity of the computing jobs both rise with the magnitude of the data input, which has an effect on the total cost.The columns labelled "No of Resources," "RAM Utilisation (%)," and "Time (ms)" show the efficiency and features of resource allocation for each scenario. For example, the computation time in the first row, which has 100 MB of data input, is 80 ms, and five resources are allocated with 70% RAM utilization.

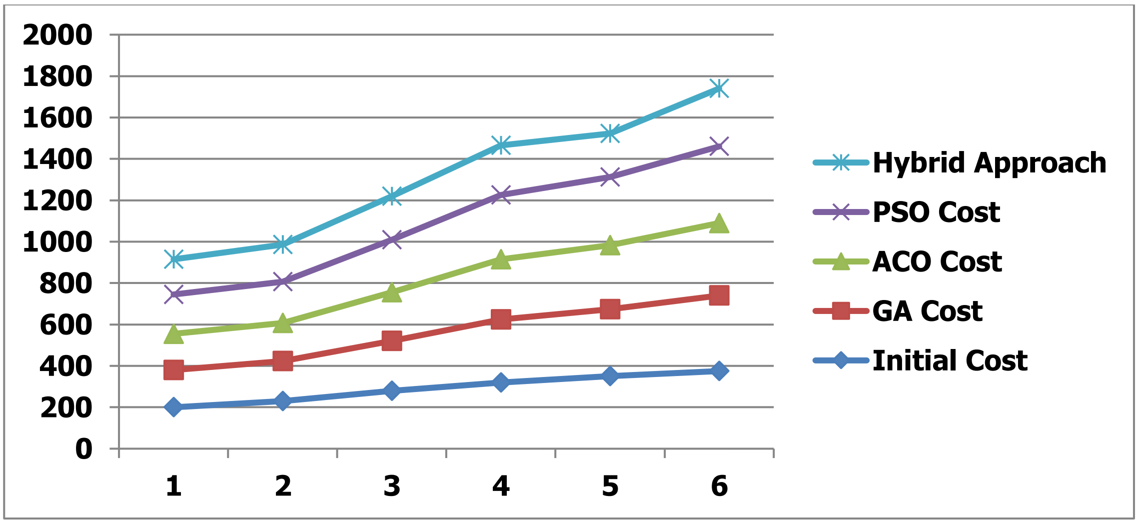

Fig 5: Comparison of Load balancing Optimization cost with different methods

The cost prior to the application of any optimisation techniques is shown in the "Initial Cost" column. The expenses spent following the implementation of GA, ACO, PSO, and a Hybrid Approach are then displayed in the corresponding columns. The Hybrid Approach stands out for its ability to consistently achieve lower costs in various settings, highlighting the value of combining diverse optimisation techniques.

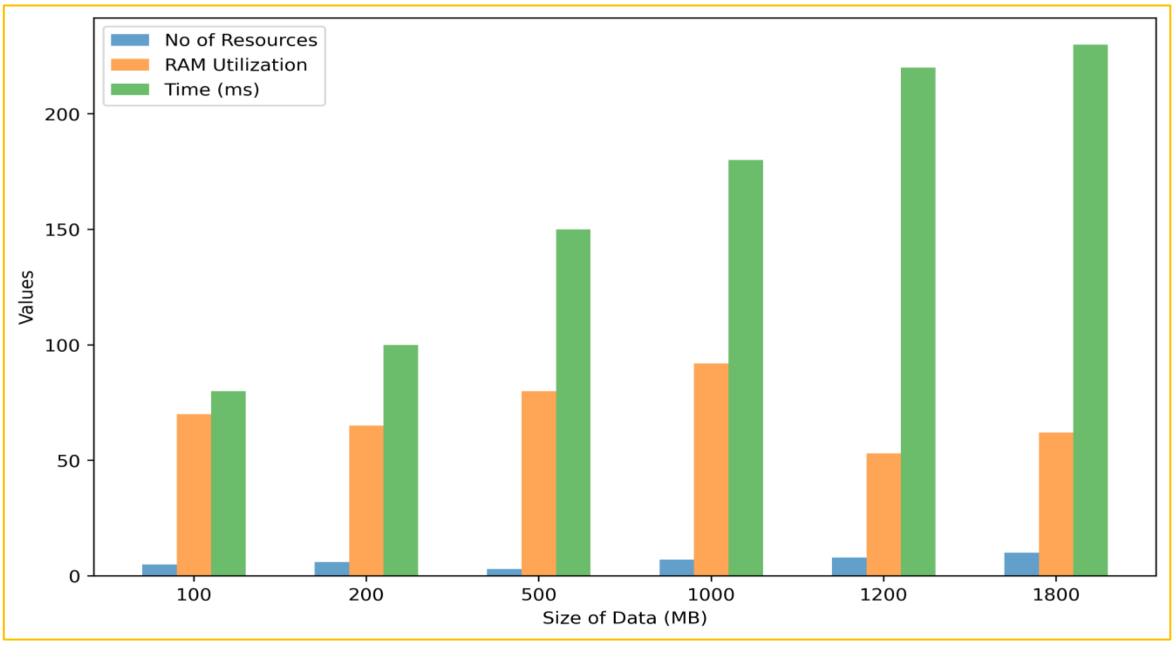

Fig 6: Comparison of Load Balancing Parameters

By examining the chart, it is clear that the Hybrid Approach performs better than individual optimisation techniques, demonstrating its versatility for a range of workloads. In some situations, GA, ACO, and PSO each have advantages; the hybridization takes advantage of these advantages to produce more economical solutions.Table 4 illustrates the complex interplay among input data size, resource distribution, and optimisation expenses within the framework of cloud computing. With this data, decision-makers can customise load balancing strategies according to workload characteristics, which will eventually optimise costs and guarantee effective resource utilisation in the ever-changing cloud computing ecosystem.

5. Conclusion

The investigation of a productive nature-inspired optimisation technique for cloud load balancing represents a major advancement in improving resource efficiency in cloud computing settings. Inspired by natural processes like Particle Swarm Optimisation (PSO), Genetic approach (GA), Ant Colony Optimisation (ACO), and their hybridization, the suggested approach aims to minimise response time and maximise throughput.The algorithm's effectiveness stems from its capacity to adjust to the dynamic nature of user requests, hence reducing imbalances brought about by variations in task distribution and the varied nature of user task demand. The technique exhibits a flexible and resilient method for attaining evenly distributed loads among virtual machines by using the advantages of ACO, GA, and PSO in a hybrid model.The study emphasises how important load balancing is in cloud computing, where effective job scheduling is an NP-hard (non-deterministic polynomial-time) problem. By offering a precise and effective mapper function, adapting dynamically to user demands, and guaranteeing optimal performance even in the face of varying workloads, the suggested algorithm solves this problem.In order to achieve Service Level Agreements (SLAs) and maintain Quality of Service (QoS), Cloud Service Providers (CSPs) must make sure that the load is evenly distributed. Efficiency and resource performance are negatively impacted by an imbalanced load. As demonstrated in this study, the algorithm adds to the CSPs' toolkit of resources to help them accomplish these goals.To put it briefly, this study presents a viable approach to cloud resource optimisation that will help create a cloud infrastructure that is more flexible and responsive. The hybrid methodology and dynamic load balancing capability of the nature-inspired algorithm make it an invaluable tool in the continuous quest to optimise cloud computing infrastructures' efficiency and performance.

References

[1] S. M. Bozorgi, M. R. Hajiabadi, A. A. R. Hosseinabadi, and A. K. Sangaiah, “Clustering based on whale optimization algorithm for IoT over wireless nodes,” Soft Computing, vol. 25, no. 7, pp. 5663–5682, 2021.

[2] A. Mosa and N. W. Paton, “Optimizing virtual machine placement for energy and SLA in clouds using utility functions,” Journal of Cloud Computing, vol. 5, no. 1, p. 17, 2016.

[3] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397–1420, 2012.

[4] Y. Ding, X. Qin, L. Liu, and T. Wang, “Energy efficient scheduling of virtual machines in cloud with deadline constraint,” Future Generation Computer Systems, vol. 50, pp. 62–74, 2015.

[5] A. Javadpour, G. Wang, S. Rezaei, and S. Chend, “Power curtailment in cloud environment utilising load balancing machine allocation,” in Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1364–1370, IEEE, Guangzhou, China, October 2018.

[6] A. K. Sangaiah, M. Y. Suraki, M. Sadeghilalimi, S. M. Bozorgi, A. A. R. Hosseinabadi, and J. Wang, “A new meta-heuristic algorithm for solving the flexible dynamic job-shop problem with parallel machines,” Symmetry (Basel), vol. 11, no. 2, 2019.

[7] A. K. Sangaiah, S. R. Ali, S. R. H. Ali et al., “Energy-aware geographic routing for real time workforce monitoring in industrial informatics,” in Proceedings of the IEEE Internet Things Journal, p. 1, NewYork, NY, USA, February 2021.

[8] S. G. Domanal and G. R. M. Reddy, “Optimal load balancing in cloud computing by efficient utilization of virtual machines,” in Proceedings of the Communication Systems and Networks (COMSNETS), 2014 Sixth International, pp. 1–4, Bangalore, India, January 2014.

[9] A. Javadpour, “Providing a way to create balance between reliability and delays in SDN networks by using the appropriate placement of controllers,” Wireless Personal Communications, vol. 110, pp. 1057–1071, 2019.

[10] A. Javadpour, G. Wang, and S. Rezaei, “Resource management in a peer to peer cloud network for IoT,” Wireless Personal Communications, vol. 115, pp. 2471–2488, 2020.

[11] A. Javadpour, N. Adelpour, G. Wang, and T. Peng, “Combing fuzzy clustering and PSO algorithms to optimize energy consumption in WSN networks,” in Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing Advanced Trusted Computing Scalable computing. Communications Cloud Big Data Computing Internet People Smart City Innovations, pp. 1371–1377, Guangzhou, China, October 2018.

[12] A. Priyadharshini and T. Nadu, “A survey on security issues and countermeasures in cloud computing storage and a tour towards multi-clouds,”

International Journal of Engineering and Technology, vol. 1, no. 2, pp. 1–10, 2013.
[13] A. Javadpour, K. Saedifar, G. Wang, and K.-C. Li, "Optimal execution strategy for large orders in big data: order type using Q-learning considerations," Wireless Personal Communications, vol. 112, pp. 123–148, 2020.
[14] A. Javadpour, "Improving resources management in network virtualization by utilizing a software-based network," Wireless Personal Communications, vol. 106, no. 2, pp. 505–519, 2019.
[15] A. M. Manasrah and H. Ba Ali, "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," Wireless Communications and Mobile Computing, vol. 2018, Article ID 1934784, 16 pages, 2018.
[16] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 305–316, 2016.
[17] S. Banerjee, M. Adhikari, and U. Biswas, "Development of a smart job allocation model for a Cloud Service Provider," in Proceedings of the Business and Information Management (ICBIM), 2014 2nd International, pp. 114–119, Durgapur, India, January 2014.
[18] A. Javadpour, "An optimize-aware target tracking method combining MAC layer and active nodes in wireless sensor networks," Wireless Personal Communications, vol. 108, pp. 711–728, 2019.
[19] A. Javadpour and H. Memarzadeh-Tehran, "A wearable medical sensor for provisional healthcare," in Proceedings of the ISPTS 2015 - 2nd International. Symposium on Physics.andTechnologyof Sensors Dive Deep Into Sensors, pp. 293–296, Pune; India, March 2015.
[20] A. Javadpour, H. Memarzadeh-Tehran, and F. Saghafi, "A temperature monitoring system incorporating an array of precision wireless thermometers," in Proceedings of the International Conference on Smart Sensors and Application (ICSSA), pp. 155–160, Kuala Lumpur, Malaysia, May 2015.
[21] F. Jafari, S. Mostafavi, K. Mizanian, and E. Jafari, "An intelligent botnet blocking approach in software defined networks using honeypots," Journal of Ambient Intelligence Humanized Computing, vol. 12, no. 2, pp. 2993–3016, 2021.
[22] A. K. Sangaiah, M. Sadeghilalimi, A. A. R. Hosseinabadi, and W. Zhang, "Energy consumption in point-coverage wireless sensor networks via bat algorithm," IEEE Access, vol. 7, pp. 180258–180269, 2019.
[23] P. Jain and S. K. Sharma, "A Systematic Review of Nature inspired Load Balancing Algorithm in Heterogeneous Cloud Computing Environment," in Proceedings of the 2017 Conference on Information and Communication Technology (CICT), pp. 1–7, Ghaziabad, India, November 2017.
[24] B. Huang, W. Wei, Y. Zhang et al., "A task assignment algorithm based on particle swarm optimization and simulated annealing in Ad-hoc mobile cloud," in Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6, October 2017.
[25] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected internet of Things," IEEE Internet of Things Journal, vol. 4, no. 2, pp. 404–413, 2017.