Minimax Theory and its Applications

Volume 11 (2026), No. 3s, 854 to 861

A Threshold-Based Hybrid GWO–PSO Framework with Trust-Aware Security for Efficient Load Balancing in Cloud Computing

Venkateswarulu K[1], Amarendra K[2]

[1,2]Department of Computer Science and Engineering, KoneruLakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
Email: kondiba9911@gmail.com, amarendra@kluniversity.in

Abstract
Cloud computing is becoming increasingly complex and is constantly evolving as it uses a variety of resources to perform tasks. The challenge lies in monitoring cloud providers and ensuring they use the appropriate amount of computer resources for each task. This study presents an intelligent workload allocation model created using a hybrid optimisation approach that utilises Grey Wolf Optimisation (GWO) and Particle Swarm Optimisation (PSO). This solution uses a nonlinear composite model to track the resources allocated to a virtual machine (VM) and adjust the thresholds that determine the VM's system state, ensuring processing continues smoothly. To speed up the process of assigning jobs and make the most of the computer resources used for this purpose, a priority-based approach will be implemented. This will enable work to be shared between computers, ensuring that the servers delivering services do not become overloaded. A trust-based virtual machine filtering provides a lightweight security-awareness layer that ensures dependable task allocation and efficient load balancing.Our experimental results show that the proposed workload distribution method surpasses existing load balancing techniques for each participating cloud service provider across all quality metrics (e.g. total make span, energy consumption, throughput and degree of distribution imbalance). This study shows that the GWO-PSO hybrid optimisation framework can provide fast, flexible load balancing solutions for cloud computing environments.

Keywords: Cloud computing, Load balancing, Grey Wolf Optimisation, Particle Swarm Optimisation

1. Introduction

Cloud computing enables businesses to use the internet to access scalable, on-demand computing resources. Cloud computing uses in Internet of Things (IoT), real-time apps and processing large amounts of data [1]. These technologies connect businesses, enabling them to make better use of their resources. This gives businesses more freedom and makes it easier and low cost to maintain the supporting infrastructure. However, the unexpected strain that cloud services place on resources has made it more difficult to monitor their performance, particularly as the number of cloud provider services has increased significantly [2]. A significant challenge for businesses using cloud-based services is uneven load distribution. Load imbalances can harm businesses in various ways, including reducing throughput, increasing energy costs and slowing response times [3].

To solve issues arising from poor resource allocation leading to excessive load imbalances, it is crucial to distribute workloads evenly across the cloud vendor's virtual machines (VMs) for optimal load distribution [4]. However, most load balancing technologies on the market today cannot quickly adapt to constantly changing workloads [5]. A company that does not use load balancing properly often wastes a lot of money. In order to make optimal use of resources, we need to develop new, intelligent and adaptable load balancing algorithms that can swiftly respond to changes in the cloud computing system.

To address the above challenges, we proposed a load balancing system that integrates both GWO and PSO methodologies, founded on threshold parameters. We require an intelligent method for optimisation and a flexible decision-making process. The proposed model's ability to generate dynamic criteria will enable real-time sorting of virtual machine states. A non-linear composite load model will be employed to illustrate the system's actual functioning. This method will also enable you to redistribute the load to prevent overloads and maintain system stability. It will feature a hybrid optimisation system that assists in determining the allocation of virtual machines (VMs) for tasks, hence maintaining balanced load distribution and enhancing response times.

The organisation of the paper as follows: In section 2 describes the various existing load balancing methods, the section 3 demonstrates the hybrid GWO-PSO load balancing model. In section 4 illustrates the performance and analysis of the proposed hybrid GWO-PSO model. Finaly section concludes that the overview of the proposed model.

2. Related work

Task-Based System Load Balancing using Particle Swarm Optimisation (TBSLB-PSO) [6] balances system load by moving only extra tasks from an overloaded VM, rather than moving the entire VM. This model also created an optimisation model that uses PSO to transfer these additional workloads to new host virtual machines. The static task scheduling technique that employs the PSO algorithm. It is based on the assumption that workloads are non-interruptible and distinct from one another. This model have improved upon the fundamental PSO algorithm by implementing a load-balancing technique [7]. A secure design with an additional security feature has been developed to address the issues of joint load balancing and compute offloading in MEC systems. Shifting MDUs between small base stations is an effective way to balance the load. To improve data security, a more advanced AES-based encryption method uses keys derived from ECG signals [8].

Author [9] addressed the issue of distributed load balancing for IoT applications. This focus on reducing missed deadlines, total deployment costs and the number of services not hosted by edge or cloud nodes within larger systems. To achieve this, we will optimise local metrics (e.g. QoS) and global metrics (e.g. total workload) to distribute IoT services effectively on edge and cloud nodes. The EPOS Fog model is designed to distribute the load evenly while keeping costs low. It uses decentralised systems comprising multiple agents to facilitate collaboration between edge and cloud nodes. Authors [10] proposed a real-time virtual machine migration technique for load balancing in fog computing. The primary objectives of load balancing algorithms are to enhance productivity, efficiency and speed. The real-time virtual machine migration method achieves cost savings that are 18% better than those of the closest data centre and provides a faster reaction time.

The authors [11] proposed resolution permits CPU issues to occur in virtual machines so that the cloud computing framework can be as trustworthy and accessible as possible. While a virtual machine is running, the CPU may malfunction. The main purpose of the proposed model is to monitor CPU usage and take appropriate action when a significant increase is detected.The authors propose an efficient FHLBA [12] to accelerate the average load, turnaround time and response time of a data centre. This method uses the firefly algorithm to determine the most effective way to distribute work among the virtual machines within the search space. The authors also examine HEPGA [13], a job allocation technique that uses Genetic Algorithms and Particle Swarm Optimisation to improve efficiency. They demonstrate how to utilise it to maximise resource utilisation when dealing with diverse work objectives and tasks.

The authors [14] presented a method for dynamically balancing the load in cloud computing. This method employs deep learning models incorporating both CNNs and RNNs to determine the load of each virtual machine. The HLFO method uses reinforcement learning to group virtual machines that are either overloaded or underutilised to optimise resource usage. Meta Reinforcement Learning Driven Hybrid Lyrebird Falcon Optimisation for Dynamic Load Balancing in Cloud Computing (Meta-RHDC) is a new approach to load balancing in the cloud [15]. The Meta-RHDC model uses recurrent and convolutional neural networks to determine how much work a virtual machine needs to perform before it can be deemed too busy or too idle.

A threshold-based load balancing [16] technique that ensures a fair distribution of workloads among nodes. The main purpose of the strategy is to prevent virtual machines (VMs) in the cloud from becoming overloaded or inactive due to incorrect task allocation when other tasks are already in progress. The threshold settings in our system ensure that VMs are used properly, preventing both overload and idleness resulting from incorrect task assignment.

3. Proposed Hybrid PSO-GWO Load Balancing Model

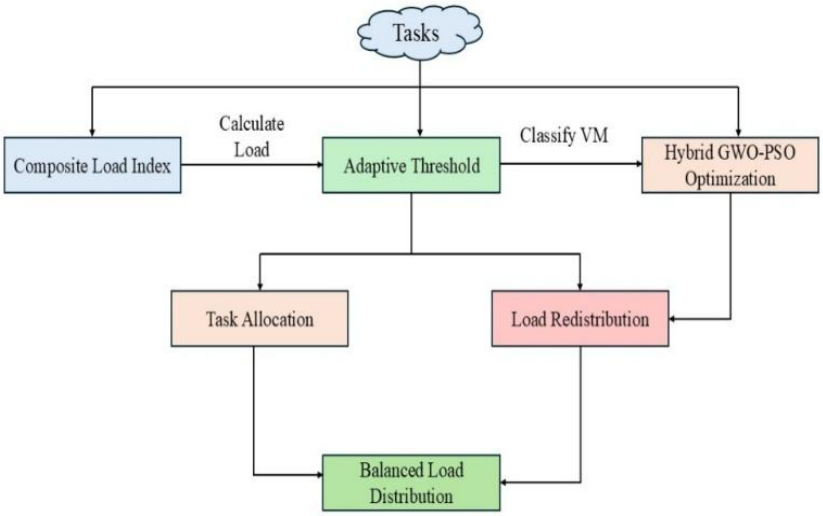


Fig. 1: Flow diagram of the proposed hybrid GWO-PSO model

The figure 1 describes the flow diagram of the proposed GWO-PSO model. A cloud with N virtual machines executing tasks at varying intervals can be envisioned. Each VM utilises resources and queues in different ways. A nonlinear composite load index has been developed to accurately reflect this evolving state.

$$L_i = \frac{\left(C_i^{\theta} + M_i^{\phi}\right)}{1 + e^{-Q_i}} + \eta \cdot \sin(Q_i)$$

Here, C_i indicates CPU utilisation, M_i Mi denotes memory consumption and Q_i represents the length of the task queue for virtual machine i. The scaling exponents ϕ and ηis indicate the interdependence of the resources and the sensitivity of the queue to fluctuations. The exponential denominator reduces the effect of queue extension, and the sinusoidal component illustrates the temporal variations in the queue, improving the accuracy and adaptability of the load representation. Unlike conventional linear aggregation models, this formulation incorporates nonlinear interactions between system resources and temporal variations in workload. This improves the precision of load predictions and facilitates better balancing decisions.

In a dynamic cloud architecture, static threshold settings may not reflect variations in workload. Dynamic thresholds, also known as self-adjusting thresholds, have been developed to enable more effective responses to such fluctuations. The following explanation outlines the higher and lower thresholds.

$$T_{high} = \mu_L + \kappa \cdot \frac{1}{N}\sum_{i=1}^{N}(L_i - \mu_L)^2$$

$$T_{low} = \mu_L - \kappa \cdot \tanh(\sigma_L)$$

whereμ_L is the average load for all virtual machines, σ_L is the load dispersion, and κ is a sensitivity coefficient that shows how flexible of the threshold. The quadratic deviation term in T_{high} makes it easier to find masses that aren't even. As things change, the hyperbolic tangent function in T_{low} always changes the lower thresholds. The adaptive criteria let the system figure out in real time if each virtual machine is overloaded, balanced, or underloaded. This makes sure that the system can handle changes in workload and that the work is divided up fairly.

This work presents a novel approach to improving the allocation of tasks between the particle swarm optimisation (PSO) and grey wolf optimisation (GWO) methods. Combining GWO and PSO into one algorithm results in a hybrid method. A perturbation term is then applied to alter the possible solutions.

$$X_i^{t+1} = \frac{X_\alpha + X_\beta}{2} + \lambda \cdot (X_\delta - X_i^t) \cdot \cos(\omega t)$$

where$X_\alpha, X_\beta, X_\delta$represent leading solutions guiding the search, and λ, ωare exploration control parameters. To further refine convergence, the velocity update is modified as:

$$V_i^{t+1} = \psi V_i^t + \rho_1 (P_i - X_i^t)^2 + \rho_2 \cdot \ln(1+| G - X_i^t |)$$
$$X_i^{t+1} = X_i^{t+1} + V_i^{t+1}$$

where$V_i$denotes velocity, P_i is the local best position, G is the global best solution, and $\psi, \rho_1, \rho_2$are adaptive coefficients controlling convergence behavior. The perturbation term makes the velocity of each candidate solution oscillate in a regulated way. This facilitates the discovery of new solutions (GWO) and building on existing ones (PSO). Combining the two approaches into one hybrid algorithm improves the search by increasing variety, speeds up convergence and reduces the likelihood of getting stuck in local optima during load balancing optimisation.

Task allocation is carried out using a priority-aware metric that evaluates the suitability of assigning task T_j to virtual machine VM_i:

$$\Omega_{ij} = \frac{L_i}{1 + e^{-P_j}} + \delta \cdot RT_i$$

where$L_i$denotes the current load of the VM, $P_j$represents task priority, and $RT_i$is the expected response time, while δ controls the influence of delay. The system is kept balanced by the fact that higher-priority jobs are done more quickly than lower-priority jobs. When a job is assigned to a virtual machine (VM), it is divided equally among several VMs and assigned to the VM with the lowest Ω_{ij}value, which speeds up response time. In order to maintain a balanced system, minimise bottlenecks and improve reliability, jobs will be transferred to a VM that has either met or exceeded its maximum load capacity, depending on the current state of the system.

Security-Aware Load Allocation

In addition to its efficient load balancing, the proposed Hybrid GWO-PSO architecture provides a lightweight, security-aware mechanism for job assignment. Before work is assigned to a virtual machine (VM), a trust score is assigned to it based on operational stability, authentication status, abnormal access records, and historical execution dependability. The cloud controller selects the resources based on the trust value, which changes dynamically.

For each virtual machine (VM_i), we assign a security confidence value (S_i) in the range [0, 1), where a higher value means a better reliability. We also use a generalised task allocation index.

$$A_i = \frac{P_t, S_i}{(L_i + 1)(1 + \lambda R_i)}$$

Where,P_t denotes task priority, L_i represents the current load of the VM, R_i is the estimated response time, and λ controls the response-time influence. During allocation, only virtual machines satisfying

$$S_i \geq S_{th}$$

These are applicable for scheduling when the minimum security level is S_{th}.Trust values are updated frequently and used to inform scheduling decisions. The security-aware filtering algorithm has a very low computational cost. This allows the suggested approach to balance workloads efficiently and limit the probability of sending jobs to potentially compromised or unreliable virtual machines, thereby improving overall cloud dependability.

4. Results and Discussion

We used CloudSim, a tool that enables model and test cloud computing systems, to test the proposed MARL–GA load balancing framework. The simulated environment comprises a data centre containing a number of hosts and 300 virtual machines (VMs) configured to perform various tasks. Each host contains six CPU cores, 16 GB of RAM, 4 TB of storage and 15 GB of bandwidth per second. The hosts also have a virtual machine monitor called Xen that works with Linux. The hardware hosts can handle 117.73 million instructions per second (MIPS). Each virtual machine has one processor, 0.5 GB of memory, 9,726 MIPS of CPU power and 1 GB/s of bandwidth. All the other tests are run on a Windows computer with an Intel Core i5 processor and 4 GB of RAM. The CPU runs at 30 GHz. The MARL-GA scheduler allocates a random computation length to each of the 500–2,000 cloudlets in the workload on the available virtual machines.

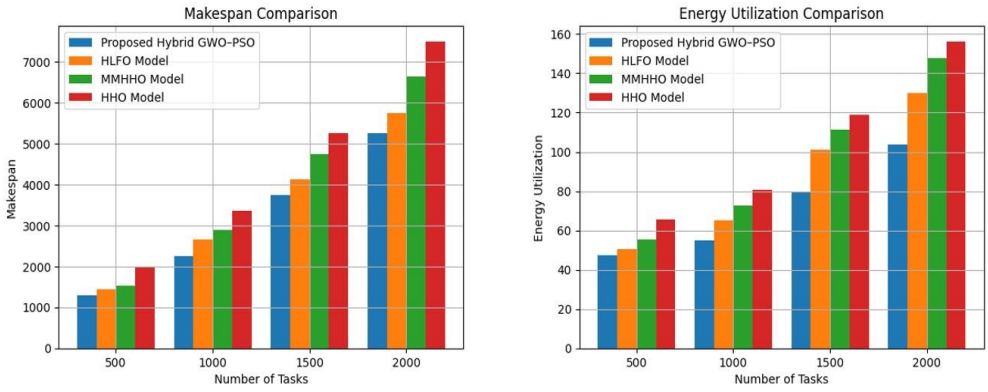


Fig. 2: Performance analysis of Makespan and Energy utilization

The model was tested using benchmark data derived from the HLFO, MMHHO and HHO models. Figure 2 demonstrates the performance analysis of Makespan and Energy utilization. When processing 500 taks, the model achieved a makespan of 1,300, outperforming the HLFO (1,450), MMHHO (1,525) and HHO (1,975) models. The total makespan for 2,000 tasks processed with the combined GWO and PSO model (5,250) was similar to that for 500 tasks (1,300). The HLFO (5,750), MMHHO (6,650) and HHO (7,500) models took much longer to complete than the hybrid GWO and PSO model. This proposed paradigm has demonstrated its ability to significantly reduce execution times by managing a greater workload within the virtual environment.

The study found that the Hybrid GWO-PSO method uses the least amount of energy overall, making it the most efficient way to get things done. For 500 tasks, the total energy usage is 47.2. In comparison, the total energy usage for HLFO, MMHHO and HHO is 50.7, 55.4 and 65.5 respectively. The Hybrid GWO-PSO Method uses 104 units of energy for a workload of 2,000 tasks. The following amounts of energy were used by the other methods tested: 130.2 for HLFO, 147.6 for MMHHO, and 156.3 for HHO. This study demonstrates that, as system demand escalates, the Hybrid GWO-PSO method facilitates ongoing resource optimisation while decreasing total energy consumption.

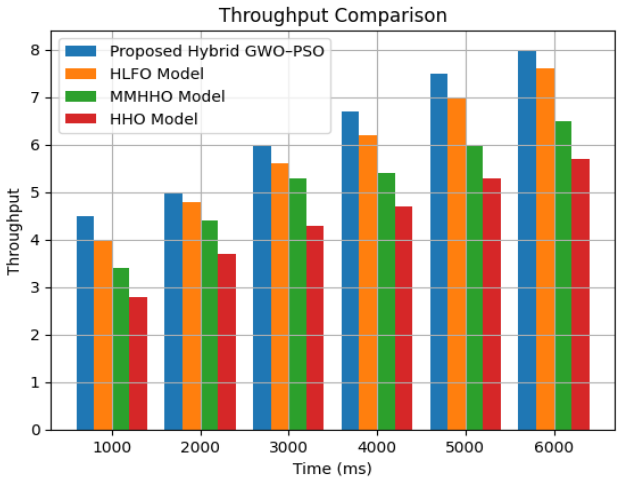


Fig. 3: Performance analysis of Throughput

Figure 3 shows that using the GWO and PSO algorithms together achieves greater performance for all other techniques. The Hybrid GWO-PSO algorithm achieved a throughput of 4.5 at a length of 1,000 milliseconds. The HLFO, MMHHO and HHO algorithms achieved throughputs of 4.0, 3.4 and 2.8 respectively. The Hybrid GWO-PSO method could handle a throughput of 8.0 at a length of six seconds (6,000 milliseconds). HLFO, MMHHO and HHO achieved throughputs of 7.6, 6.5 and 5.7 respectively. This data indicates that using both the GWO and PSO algorithms together always finishes jobs faster. They will continue to perform well regardless of how long it takes to complete the task.

Table 1: Degree Imbalance

Tasks	Before Load Balance	After Load Balance
500	65.7	19.5
1000	75	21
1500	86	25.5
2000	105.5	38

Table 1 shows that the new load balancing technique reduces inaccuracy (imbalance) in the workload between physical servers handling these workloads. For example, before implementing this strategy, there were 65.7 units of imbalance out of 500 jobs. After utilising it, this figure fell to 19.5. This demonstrates that the new load balancing method changes how the load is distributed among the physical servers allocated jobs. In a scenario involving 2,000 total workloads, the imbalance decreased from 105.5 units to just 38 units after implementing the new load balancing approach. The proposed load balancing method enables the various virtual machines (VMs) in the system to share system tasks equitably. This reduces the likelihood of some virtual machine resources becoming overloaded due to excessive workloads.

5. Conclusion

A threshold load balancing system based on a hybrid GWO-PSO has been developed that uses PSO's ability to quickly evaluate solutions and GWO's capacity to identify the optimal solution. This strategy employs the most effective methods for categorising things into adaptive thresholds. The new hybrid GWO/PSO model fixed problems with both types of method (e.g. determining the system's state and making dynamic decisions about loading), it has also improved existing methods (e.g. reducing makespan, energy use and throughput, and improving load balancing) to optimise system performance, scalability and reliability.The use of trust-based virtual machine evaluation, which improves scheduling reliability and reduces the number of resources allocated to the work with the least amount of overhead, is another benefit of the load balancing strategy we suggested. Future endeavours will likely integrate security-oriented load balancing methodologies to implement this concept in distributed cloud computing systems capable of real-time data processing.

Declarations

Ethics approval and consent to participate

Not applicable

Acknowledgment

Not applicable

Availability of data and material

Not applicable

Competing interests

Authors do not have any Competing interests.

Funding

NA

Consent for Publication

The authors declare that they have all participated in the preparation of this manuscript and have approved the final version for submission. All authors consent to the publication of this work in a Springer journal.

The authors confirm that the manuscript has not been previously published and is not under consideration for publication elsewhere. All necessary permissions for reproduction of any copyrighted material have been obtained.

Authors' contributions

Venkateswarulu K contributed to data collection, implementation of the proposed model, experimental evaluation, coding, result interpretation, and drafting of the manuscript.

Amarendra K conceptualized the research idea, supervised the study, and provided critical guidance on methodology and manuscript review.

References

1. Nazir R, Ahmed Z, Ahmad Z, Shaikh NN, Laghari AA, Kumar K. Cloud Computing Applications: A Review. EAI Endorsed Transactions on Cloud Systems. 2020 Jan 1;6(17).

2. Iosup A, Ostermann S, Yigitbasi MN, Prodan R, Fahringer T, Epema D. Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Transactions on Parallel and Distributed systems. 2011 Feb 24;22(6):931-45.

3. Ghomi EJ, Rahmani AM, Qader NN. Load-balancing algorithms in cloud computing: A survey. Journal of Network and Computer Applications. 2017 Jun 15;88:50-71.

4. Prity FS. Nature-inspired optimization algorithms for enhanced load balancing in cloud computing: a comprehensive review with taxonomy, comparative analysis, and future trends. Swarm and Evolutionary Computation. 2025 Aug 1;97:102053.

5. Diwaker C, Miglani N. A Two-Decade Analysis of Load Balancing in Cloud Computing: Implications for Educational System and Future Directions. Journal of Computer Assisted Learning. 2025 Jun;41(3):e70042.

6. Ramezani F, Lu J, Hussain FK. Task-based system load balancing in cloud computing using particle swarm optimization. International journal of parallel programming. 2014 Oct;42(5):739-54.

7. Ebadifard F, Babamir SM. A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment. Concurrency and Computation: Practice and Experience. 2018 Jun 25;30(12):e4368.

8. Zhang WZ, Elgendy IA, Hammad M, Iliyasu AM, Du X, Guizani M, Abd El-Latif AA. Secure and optimized load balancing for multitier IoT and edge-cloud computing systems. IEEE Internet of Things Journal. 2020 Dec 3;8(10):8119-32.

9. Nezami Z, Zamanifar K, Djemame K, Pournaras E. Decentralized edge-to-cloud load balancing: Service placement for the Internet of Things. Ieee Access. 2021 Apr 28;9:64983-5000.

10. Yu D, Ma Z, Wang R. Efficient smart grid load balancing via fog and cloud computing. Mathematical Problems in Engineering. 2022;2022(1):3151249.

11. Ramya K, Ayothi S. Hybrid dingo and whale optimization algorithm-based optimal load balancing for cloud computing environment. Transactions on Emerging Telecommunications Technologies. 2023 May;34(5):e4760.

12. Srivastava A, Kumar N. An efficient firefly and honeybee based load balancing mechanism in cloud infrastructure. Cluster Computing. 2024 Jun;27(3):2805-27.

13. Mikram H, El Kafhali S, Saadi Y. HEPGA: A new effective hybrid algorithm for scientific workflow scheduling in cloud computing environment. Simulation modelling practice and theory. 2024 Jan 1;130:102864.

14. Khan AR. Dynamic load balancing in cloud computing: optimized RL-based clustering with multi-objective optimized task scheduling. Processes. 2024 Mar 4;12(3):519.

15. Krishna MS, Vali DK. Meta-RHDC: meta reinforcement learning driven hybrid lyrebird falcon optimization for dynamic load balancing in cloud computing. IEEE Access. 2025 Feb 24;13:36550-74.

16. Chowdhury S, Katangur A. Optimizing Cloud Computing Performance Through Integration of a Threshold-Based Load Balancing Algorithm With Multiple Service Broker Policies. IEEE Transactions on Cloud Computing. 2025 Apr 24.