RESEARCH PAPER

Qsas: A Novel Quantum State-Based Adaptive Scheduling Algorithm for Cloud Resource Optimization

G. Abarna[1], Dr. S. Thilagavathi[2]

[1] Research Scholar, Sri Krishna Adhithiya College of Arts and Science. Email: abarnag@skasc.ac.in

[2] Department of Computer Science, Sri Krishna Adhithiya College of Arts and Science.
Email: thilagavathis@skacas.ac.in

Received: 2nd Mar, 2026 | **Revised:** 14th Mar, 2026 | **Accepted:** 4th Apr, 2026 | **Available Online:** 20th Apr, 2026

ABSTRACT

Cloud computing environments demand increasingly sophisticated scheduling solutions to optimize resource allocation and operational efficiency. In this paper, Quantum State-based Adaptive Scheduling (QSAS), a novel algorithm is introduced which utilizes the principles of quantum mechanics to enhance cloud scheduling processes. This method maps cloud scheduling problems onto quantum states and uses adaptive optimization techniques that adjust dynamically based on the real-time cloud resource demands. Compared to the traditional methods, this unique approach enables QSAS to explore complex scheduling solutions with higher efficiency. This algorithm integrates adaptive annealing schedules to fine-tune the optimization process for varying cloud workloads. Performance and security evaluations of QSAS are conducted through extensive simulations, demonstrating its ability to scale efficiently across large cloud environments. The results show the potential of quantum computing to revolutionize cloud scheduling with QSAS leading the way in providing a scalable, secure, and practical solution for future cloud infrastructure.

Keywords: Quantum State-based Adaptive Scheduling (QSAS), Cloud computing, Cloud resource, Optimization, Resource allocation.

How to cite this article: Abarna G, Thilagavathi S. Qsas: A Novel Quantum State-Based Adaptive Scheduling Algorithm for Cloud Resource Optimization. Int J Drug Deliv Technol. 2026;16(35s):246-257. DOI: 10.25258/ijddt.16.35s.26

Source of support: Nil.

Conflict of interest: The authors declare no conflict of interest.

I INTRODUCTION

Cloud computing has been the foundation of modern digital infrastructure, which allows for the scalable and provisioning of resources for various applications in healthcare, finance, artificial intelligence, and big data analytics. However, efficient computational task scheduling within the resource utilization optimization remains one of the critical challenges in dynamic workload, energy constraint, and Quality of Srvice (QoS) requirements from [1], [2]. Traditional scheduling approaches such as heuristic and metaheuristic algorithms have been widely investigated for better efficiency [3], [4], but these approaches usually lag from the scalability issues in large-scale cloud environments.

To deal with such challenges, various optimization techniques have been proposed. With the powerful exploration capabilities, Genetic algorithms had been applied in task scheduling problems [5]. On the other hand, swarm intelligence-based methods such as Particle Swarm Optimization (PSO) and Cuckoo search prove to achieve better load balance and energy efficiency [6], [7]. Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) are the few among the multi-objective optimization approaches applied to decision-making in the trade-off between execution time and resource consumption [8], [9]. More recently, adaptive techniques have emerged where the task scheduling is readjusted dynamically in accordance with the changing workload in the cloud [10], [11].

Adaptive scheduling has been considered significantly, given its ability in responding to the real-time workload variations and optimizing cloud resource allocation [12], [13]. The methods of reinforcement learning and Self-Learning Automata (SLA) have further improved the efficiency in cloud scheduling [14], [15]. Moreover, secure scheduling mechanisms are bound to become increasingly important because the cloud computing environments need strong security frameworks to protect the sensitive data [16]. More recently, some hybrid scheduling approaches combines different techniques in a single framework like Artificial Bee Colony (ABC) algorithms with Symbiotic Organism Search (SOS). These approaches have been presented to enhance both performance and security [17], [18].

Despite these advances, there are fundamental limitations in efficiently handling the classical complex and dynamic scheduling scenarios using classical algorithms. Quantum computing has emerged as a new paradigm for solving the complex optimization problems by using the principles of superposition and entanglement to search the vast solution spaces more efficiently [19], [20]. Inspired by such principles, Quantum State-based Adaptive Scheduling (QSAS), an innovative quantum-inspired scheduling framework is introduced where the cloud scheduling problems are modeled as quantum states combine with adaptive annealing optimization dynamically optimizes the cloud resources.

Contributions:

This presented method QSAS maps the cloud scheduling problem to the quantum state and harvests the adaptive optimization methodology for better decision making. It incorporates an adaptive annealing schedule in the optimization process to adjust it dynamically for the dynamic cloud workloads. A cloud environment is simulated in large-scale simulations to evaluate QSAS through extensive simulations, which outperform the classical approaches in scheduling efficiency, resource utilization, and security.

This paper is organized as follows: Section 2 discusses the related work on cloud scheduling and quantum-inspired optimization; Section 3 details the QSAS framework and algorithmic design; Section 4 presents the experimental setup and evaluation metrics; Section 5 analyzes the results by comparing QSAS with the existing scheduling methods and finally, Section 6 concludes the paper with future research directions.

II BACKGROUND STUDY

Chen et al. [21] discussed a resource-constrained profit optimization method for task scheduling in edge cloud environments. These authors introduced a profit-driven approach that balanced the resource utilization and operational costs. The authors reviewed existing task scheduling methods and identified the inefficiencies in cost management. While considering the resource constraints, a theoretical model was developed to optimize the task allocation. This research aimed to improve scheduling efficiency and profit maximization. It led the way for incorporating the economic aspects into resource management. The proposed model had been extended to optimize large-scale cloud systems.

Chen et al. [22] proposed a Whale Optimization Algorithm (WOA)-based task scheduling strategy for cloud computing. These authors analyzed the existent metaheuristic algorithms and found some limitations in convergence speed and task distribution. The study formulated an optimization framework by incorporating WOA for better resource management. The goal was to improve the efficiency of scheduling. By comparing the WOA-based strategy with other heuristic methods, this research contributes a comparative evaluation. The results showed that the intelligent algorithms had substantially enhanced the task allocation process. Future studies combined the hybrid approaches for further optimization.

Dewangan et al. [23] introduced a cloud resource optimization system based on time and cost constraints. It highlighted the inefficiencies of traditional scheduling algorithms in handling cost and time trade-offs. These authors proposed a mathematical model that minimized the resource wastage and maximized the efficiency. The research aimed to develop an adaptive system for optimal resource utilization. The framework was validated using real-time cloud scenarios. The findings suggested that balancing time and cost had significantly improved cloud performance. The study provided a foundation for integrating Artificial Intelligence (AI)-based scheduling methods.

Garg & Jindal [24] explored the predictive Virtual Machine (VM) consolidation for resource optimization in cloud computing. This study reviewed the existing VM consolidation techniques and highlighted issues in predicting workload variations. A predictive model was developed using Machine Learning (ML) to enhance VM allocation. The research aimed to reduce energy consumption and improve computational efficiency. The proposed approach showed the promising results in minimizing underutilized resources. It contributed to advanced energy-aware cloud scheduling mechanisms. Further research explores the Deep Learning (DL)-based predictive models for cloud optimization.

Junaid et al. [25] proposed an optimized load balancing approach for cloud environments. These authors analyzed the traditional load balancing strategies and identified the bottlenecks in handling dynamic workloads. The study developed an intelligent model to distribute the tasks efficiently across multiple servers. The objective was to enhance scalability and response time in cloud systems. This research incorporated mathematical optimization techniques to improve the system performance. The results indicated a substantial reduction in load imbalance. Thus, this study formed the basis for future adaptive load-balancing mechanisms.

Kasyap et al. [26] proposed the ML-based resource management of cloud computing. The authors had analyzed the various decision-making frameworks and identified the predictive resource allocation gaps. This work developed a learning-based model for improving task scheduling efficiency. The objective of this approach was to provide an adaptive mechanism for resource provisioning. The findings were demonstrated to handle the workload fluctuations in an efficient manner, as indicated by the improved efficiency in the approach. The Artificial Intelligence-driven decision-making was highlighted. Future work may also incorporate deep reinforcement learning for further optimization.

Lahande et al. [27] investigated the Reinforcement Learning (RL) for cloud resource utilization and load balancing. This study reviewed the existing applications of RL in cloud computing and pointed out the inefficiencies in adaptive load distribution. A Reinforcement Learning-based model was proposed that dynamically adjusted the resource allocation in accordance with the workload patterns. The objective was to enhance resource efficiency and minimize energy consumption. The findings proved the effectiveness of RL in real-time cloud optimization. The study contributed to AI-driven cloud resource management. Further, fFuture research explores the hybrid AI techniques for more efficiency gains.

Lattuada et al. [28] analyzed the existing optimization methods based on Spark and pointed out the challenges in handling the large-scale data workloads. While reducing the resource overhead, a model was introduced to enhance the scheduling efficiency. The aim of this study was to improve the strategies for resource allocation in distributed computing. The findings suggested that intelligent scheduling significantly improved the cloud performance. This work gave the foundation of big data application optimization in cloud environments.

Prasad et al. [29] analyzed the efficient resource utilization in Internet of Things (IoT) and cloud computing environments. These authors study reviewed the existing IoT-cloud integration frameworks and identified the inefficiencies in resource distribution. An intelligent scheduling mechanism was proposed for the optimization of IoT-cloud interactions. This research aimed in improving the speed of task execution and reduction in energy consumption. This approach demonstrated the improved efficiency in handling distributed workloads. Therefore, the results were relevant for the development of scalable cloud-based IoT frameworks. Future research could concentrate on the AI-driven methods for further optimization.

Qiu et al. [30] investigated the intelligent security and optimization in edge or fog computing. It identified the security vulnerabilities and resource management issues in edge environments. The authors proposed a security-aware optimization framework for workload distribution and threat mitigation. The objective was to improve the security without sacrificing performance. The3 results showed that intelligent security mechanisms enhanced the efficiency of edge computing. This research contributed to secure and optimized cloud-edge integration. Future work could focus on implementing blockchain to heighten the security aspects.

Table 1: Comparison of diverse Methodologies and Optimization Strategies in Cloud Resource Management

Reference	**Methodology**	**Optimization Approach**	**Unique Contribution**
Sangaiah et al. [33]	Heuristic algorithms for IoT resource management	Heuristic-based scheduling	Optimized IoT resource allocation for real-time applications
Sharma &	Ant Colony	Bio-inspired	Improved

Garg [34]	Optimization (ACO)	optimization	Quality of Service (QoS) for cloud scheduling using ACO
Shukur et al. [35]	VM allocation strategies	Virtualization techniques	Enhanced resource utilization using VM allocation policies
Suryavanshi & Chawla [36]	AI-based networking optimization	Intelligent cloud network design	Optimized cloud infrastructure for efficient Large Language Model (LLM) deployment
Ugbebor [37]	AI and ML-based resource management	Intelligent cloud resource allocation	Bridging the technology gap for small businesses using cloud AI solutions
Xiong & Yang [38]	Multi-layered resource optimization	Hybrid cloud-fog-edge architecture	Improved deployment strategies for Cyber-Physical Distributed System (CPDS) using edge computing

Table 1 represents the comparative analysis of the various optimization techniques implemented in cloud resources within different researches. The studied papers have quite unique methodologies, approaches, optimization, and applicability in rthe eal-world applications related to IoT, AI-driven cloud networking, virtualization, and edge computing.

Ramamoorthi et al. [31] introduced an AI-driven cloud resource optimization framework for real-time allocation. The literature review identified the limitations in traditional cloud resource management techniques. This study developed an AI-based decision-making model for the dynamic resource provisioning. The objective was to enhance the real-time resource allocation efficiency. The proposed model showed improved accuracy in predicting resource demands. This research contributed to the developing field of AI-based cloud computing solutions. Future work could integrate the hybrid AI techniques for further advancements.

Reddy and Reddy [32] presented the multi-objective scheduling framework for cloud resource utilization. The literature reviewed stated that there were several performance metrics on balancing the multiple performance metrics in cloud environment. The study proposed a scheduling algorithm that aimed at finding an optimal task allocation that satisfied both energy efficiency and cost. The objective was to evolve towards a more effective distribution of resources. The results had shown the greater improvements in handling workload and operational costs. This study was useful in the development of energy-efficient cloud computing solutions. Future research could use DL techniques to improve the scheduling performance.

III .PROPOSED METHODOLOGY

Quantum State-based Adaptive Scheduling (QSAS) is developed to optimize the allocation of cloud resources by using quantum principles in dynamic scheduling. It enhances resource utilization by adapting to the workload changes in real time. Thus, reducing latency and enhancing efficiency. The algorithm functions as a quantum state to predict the forecasts for task operations along with task load balancing. While consuming minimum energy and creating the most cost-efficient operations, this system reaches the maximum throughput. Traditional scheduling techniques become ineffective for difficult cloud infrastructure, yet QSAS provides suitable solutions.

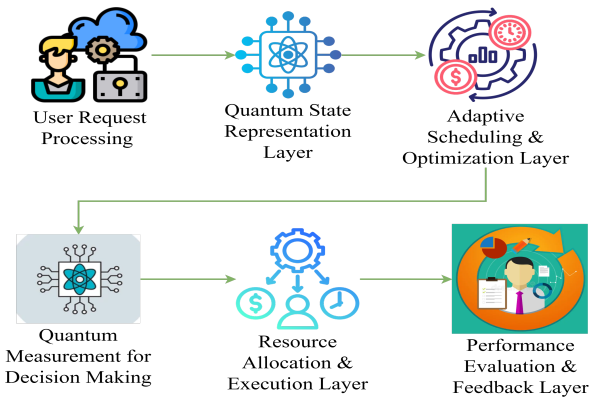

Figure 1: Overall Architecture

The multilayer architectural design of QSAS in Figure 1 provides the optimized resource allocation function. The initial layer operating from user handles request processing that involves task identification by priority before examining the execution time requirements and resource specifications. During the probabilistic scheduling, tasks are transformed into a quantum superposition state at the quantum state representation layer within the QSAS framework. The Adaptive Scheduling & Optimization layer makes the real-time system status-driven adaptive switches between such paired tasks. Through the quantum measurement, the quantum state finds its most suitable position for decision making. It selects the task-resource allocations that attain QoS prerequisites effectively. The resource allocation & execution layer completes the task assignment operations and executes the real-time monitoring processes to achieve QoS requirements. The last layer, performance evaluation & feedback layer functions to analyze the system efficiency together with resource utilization and energy consumption before making the changes to present scheduling strategies for continuous optimization purposes. The proposed system optimizes the cloud systems by enhancing flexibility as well as resource management and operational performance.

3.1 Quantum State-based Adaptive Scheduling (QSAS)

Quantum State-based Adaptive Scheduling tool operates as a contemporary optimization solution which uses quantum computing methods to manage the cloud tasks through platform-based approaches. Due to traditional methods, cloud management systems experience resource inefficiency and delay problems in scheduling. The scheduling tool QSAS evaluates numerous options directly on its quantum state-based resource and task models. To undergo the real-time adjustments of the distributed resources related to the measurement data, system metrics is exploited by the system for obtaining optimal performance by dynamically reallocating resources. The quantum measurement establishes resource distribution patterns that perform as the distribution methods minimize delay time for peak speed discovery and complete system scalability. This scheduling is essential for huge cloud platforms and data centers because it offers the processing solutions for handling large AI technology–enabled data needs. The online systems achieve crucial time-sensitive improvements through the implementation of this software application. It generates operations tolerance along with the affordable cloud management solutions because it uses quantum entanglement and superposition principles that exist within its system framework. The scheduling advantages in QSAS result in modern cloud-based systems finding the best solution through its exceptional performance during the complex system management operations.

$$|\psi\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}|T_iR_j\rangle \text{ ----------------------- } (1)$$

In equation (1), quantum state |ψ⟩ uses basis states$|T_iR_j\rangle$, with indexes T_i and R_j to describe sets that apply time slots $\sum_{i=1}^{n}\sum_{j=1}^{m}$ and resource units α_{ij} in analyzed quantum systems. A system in its specific state enables the definition of probabilities for the measurement results using its probability amplitudes. Quantum mechanics professionals and quantum information specialists use one standard system to gather the superposition and entangled states found in multi-dimensional Hilbert spaces.

$$\alpha_{ij} = f(A_j, P_i, N_{ij}) \text{ ----------------------- } (2)$$

Equation (2) establishes the coefficient α_{ij} as a function of three parameters A_j , P_i and N_{ij} . The mathematical elements A_j represent the characteristics from resource unit R_j along with the properties of time slot T_i being expressed through P_i . The notation N_{ij} represents all the interactive and dependency elements between T_i and R_j that include noise source and reliability of the connection. The probability amplitude of basis states in the superposition is calculated by this function which affects the total behavior of the quantum system.

$$\left(T_iR_j\right) = \arg{}_{i,j}^{max}|\alpha_{ij}|^2 \text{ ----------------------- } (3)$$

The optimization criteria presented in equation (3) provides the most probable state $(T_i R_j)$ when solving the quantum superposition problems. The algorithm selects pair (i, j) results in the maximum value of probability amplitude squared $|\alpha_{ij}|^2$, because it represents the measurement likelihood in that state. The decision process uses this approach to select the best configuration when allocating the resources in quantum networks to achieve maximum throughput. It determines the dominant $(T_i R_j)$ state which governs the superposition ($|\psi\rangle$).

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} (C_{ij} - \lambda |\alpha_{ij}|^2) \text{ ---------------------- (4)}$$

Equation (4) serves as an optimization method which finds the lowest possible total cost value for all (i, j) relationships. The cost parameter C_{ij} comes forth as the selection cost metric for pair $(T_i R_j)$ but $|\alpha_{ij}|^2$ denotes the probability likelihood. The parameter λ balances the influence of probability amplitudes on the total cost. It assists in choosing optimal state groups which decrease the costs while maximizing the efficiency within quantum and wireless communication systems.

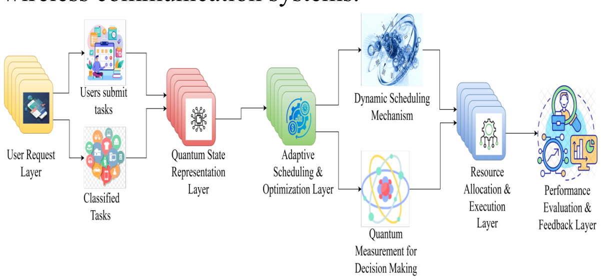

Figure 2: Architecture of Quantum State-based Adaptive Scheduling (QSAS)

Figure 2 presents a multi-layered framework which implements the quantum-inspired decision-making procedures for performing resources allocation and task scheduling operations. User Request Layer functions as the foundational lowest layer to process the user demands into different classification group tasks. A transition from classification brings classified tasks to the Quantum State Representation Layer to convert these tasks into quantum states for scheduling probability representation. The Adaptive Scheduling and Optimization Layer achieve quantum principle guided optimal scheduling by using optimization approaches. The Dynamic Scheduling Mechanism together with Quantum Measurement for Decision Making enables the selection process of optimal task-resource partnerships. Through the Resource Allocation & Execution Layer, the selected resources achieve efficient distribution. The Performance Evaluation and Feedback Layer provide the assessment of scheduling processes alongside the distribution methods to identify the areas for improvement. This method develops the enhanced operational abilities that increase the performance of the system execution as well as lowers the latency impact accompanied with optimizing the acute resource allocation specifically for cloud infrastructure.

Algorithm 1: Quantum State-based Adaptive Scheduling (QSAS)

Step 1: User Request Processing

Accept the requests from users containing computational tasks.

Classify the tasks by priority, execution time, and resource needs.

Create a queue for task scheduling.

Step 2: Quantum State Representation

Encode each task-resource pair $(T_i R_j)$ into a quantum superposition state:

$$|\psi\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} |T_i R_j\rangle$$

Compute probability amplitudes α_{ij} using a function: $\alpha_{ij} = f(A_j, P_i, N_{ij})$

Where, A_j is resource availability, P_i is task priority, and N_{ij} represents the network conditions.

Step 3: Adaptive Scheduling & Optimization

Find the most probable task-resource pair using: $(T_i R_j) = \arg \max_{i,j} |\alpha_{ij}|^2$

Dynamically apply a scheduling mechanism re-routing resources according to changes within the system pertaining to load and resources in real-time.

Step 4: Quantum Measurement for Decision Making:

Perform quantum measurement on the superposition state to collapse to the optimal task-resource pair.

Dynamical update the decisions of scheduling tasks when new tasks arrive or the resource availability changes.

Step 5: Resource Allocation & Execution

The selected tasks are allocated to the respective cloud resources.

Execution is monitored to ensure the QoS requirements are met.

Step 6: Performance Evaluation & Feedback

Evaluate the performance based on execution time, resource utilization, and energy efficiency.

Update the probability amplitudes α_{ij} to optimize future scheduling based on feedback

Repeat for the continuous optimization of cloud resources.

Algorithm 1 describes the QSAS algorithm for cloud resource allocation optimization by means of quantum-inspired decision-making. First, the user's request processing has to be realized when the arriving computational tasks are analyzed with respect to priority, execution time, and required resources. It determines the formation of a scheduling queue. In the quantum state representation, the encoding of task-resource pairs is into a quantum superposition state with probability amplitudes. It depends on resource availability, task priority, and network conditions. Adaptive Scheduling and Optimization, a quantum-inspired selection mechanism dynamically readjusts the scheduling in real time based on system load by selecting the most probable task-resource pair. Quantum Measurement for decision making, the collapsing of the quantum state determines the optimal task-resource pair, ensuring that the scheduling decisions adapt to the changing conditions. Resource Allocation and Execution module ensures that the tasks are effectively allocated to cloud resources while preserving the standards of QoS. Finally, in the Performance Evaluation & Feedback phase, probability amplitudes are updated in accordance with the efficiency of execution for the optimization of cloud resources continuously. The proposed approach enhances efficiency, adaptability, and resource utilization in dynamic cloud environments.

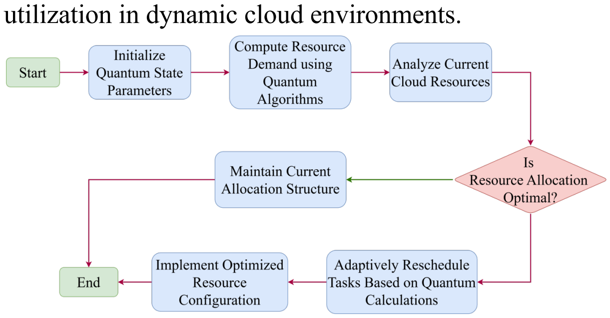

Figure 3: Flowchart of Quantum State-based Adaptive Scheduling (QSAS)

Figure 3 illustrates a quantum-based resource allocation in cloud computing. First, it initializes the quantum state parameters. Then, it calculates the resource demand by using quantum algorithms. Now, the system analyzes the current cloud resources, checking if the allocation is optimal. If the allocation is already optimal, it maintains the current structure and loops back. Otherwise, it reschedules the tasks based on quantum calculation and executes an optimized resource configuration. Finally, the process updates the allocation or terminate. In this flowchart, the role of quantum computing in dynamically optimizing the management of cloud resources is emphasized.

VI RESULTS AND DISCUSSION

Cloud resource management becomes an important aspect in modern computing, where the efficiency is measured through the multiple performance metrics. In this paper, four algorithms such as WOA, ACO, VM, and QSAS are compared based on latency, storage overhead, energy consumption, and throughput. Lower latency and energy consumption with reduced storage overhead, contribute to better algorithm efficiency. A higher throughput further improves the performance by increasing the data processing speed. By comparing these measures, this study signifies QSAS as the most optimal algorithm, whereas VM generally shows low performance in terms of efficiency.

4.1 Latency

Latency is the time taken from the initiation of a request until the first response is received. It measures the delay in processing tasks or transmitting the data within a system.

$$Latency = Transmission\,Time + Queueing\,Time + Processing \text{ ------------ } (5)$$

In equation (5), $Transmission\,Time$ refers to the time required for the transmission of data in the network. $Queueing\,Time$ indicates the time spent waiting in a queue before being processed. $Processing$ is the time required for processing the task or requesting.

4.2 Storage Overhead

The required additional storage from encryption is shown by the overhead storage. It considers any extra information, padding, or encryption-related data included into the original file.

$$Storage\ overhead = \frac{Encrypted\ Data\ size - Original\ data\ size}{Original\ data\ size} \times 100 \text{ ----------- } (6)$$

$Encrypted\ Data\ Size$ in equation (6) refers to the size of the data after encryption. $Original\ data\ size$ is the original size of the data.

4.3 Energy Consumption (J/MB)

Energy consumption gauges either during encryption or decryption, the extent of energy consumed to process a given volume of data.

$$E_{energy} = \frac{Energy\ per\ operation \times D_{data}}{Data\ size} \text{ ------------ (7)}$$

In equation (7), energy per operation is Joules equivalent of the energy used for one encryption or decryption cycle. Data size expressed in MB is the extent of data being encrypted or decrypted. E_{energy} is the data consumption expressed per MB of size.

4.4 Throughput (MB/s or GB/s)

Throughput is the capacity of data processing within a certain time span. It depends on the encryption or decryption times for data.

$$Throughput = \frac{Data\ Size}{T_{total}} \text{ ---------- (8)}$$

Data Size in equation (8) is the processed data quantity expressed in MB or GB. Including durations for encryption and decryption, T_{total} is the total time spent data processing.

Table 2: Performance comparison of Cloud resource using various algorithms

Algorithms/ Metrics	Latency (ms)	Storage Overhead %	Energy Consumption (J/MB)	Throughput MB/s
WOA [40]	50	10	0.05	5
ACO [41]	70	15	0.1	3
VM [42]	100	20	0.15	4
QSAS (Proposed)	30	5	0.02	6

Table 2 gives a performance comparison between the cloud resource management algorithms such as WOA, ACO, VM, and QSAS regarding major efficiency metrics. The presented QSAS shows the best overall performance with the lowest latency of 30 ms, minimum storage overhead of 5%, the least energy consumption of 0.02 J/MB, and the highest throughput of 6 MB/s. On the other hand, VM shows the lowest performance with the highest latency at 100 ms, the most storage overhead at 20%, and the highest energy consumption at 0.15 J/MB, although its throughput is moderate at 4 MB/s. The method WOA has better latency and is more energy efficient than ACO. This comparison shows QSAS as the most efficient algorithm. Hence, it seems to be the best choice for cloud resource management.

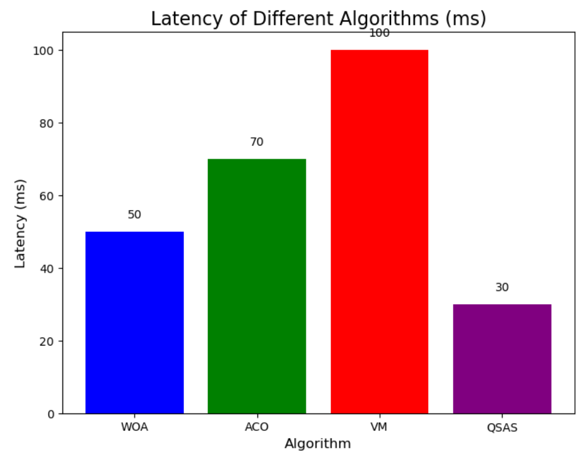

Figure 4: Latency comparison chart

Figure 4 gives a comparison of the latency in milliseconds for four different algorithms such as WOA, ACO, VM and QSAS. Latency represents a delay in processing, where the lower values indicate better performance. Thus, QSAS had the least latency of 30 ms; WOA is just below ACO at 50 ms, while ACO experienced 70 ms latency. The algorithm VM had a higher latency of 100 ms, making it much slower than others. This graph suggests that QSAS is best in terms of latency and VM has the lowest performance.

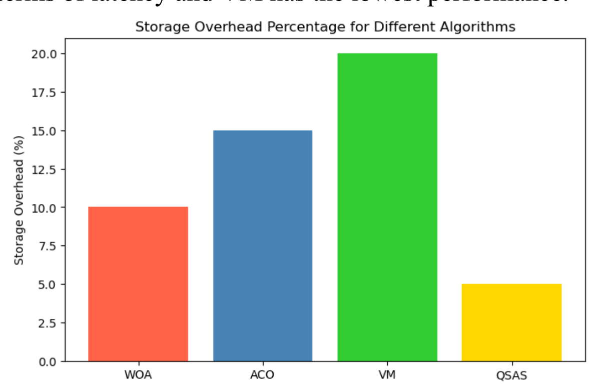

Figure 5: Storage Overhead comparison chart

Figure 5 shows the storage overhead percentage for four different algorithms such as WOA, ACO, VM, and QSAS. Storage overhead is the extra storage required beyond the actual data; the lesser the percentage, the more efficient it is. Among these, QSAS has the least storage overhead of about 5% and exhibits as the most efficient. Further, WOA shows about 10% storage while ACO has the higher overhead of about 15%. The VM algorithm has the highest storage overhead at 20%, which means it uses the additional storage. Therefore, this comparison reveals that QSAS is most efficient in storage, whereas VM is the least efficient.

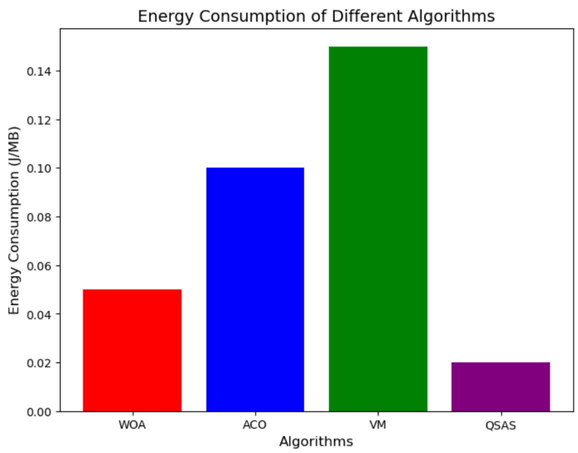

Figure 6: Energy Consumption comparison chart

Figure 6 represents the energy consumption in terms of Joules per Megabyte for the four algorithms such as WOA, ACO, VM, and QSAS. Energy consumption is one important aspect of the efficiency of an algorithm; the smaller the value, better is the energy efficiency. The proposed method QSAS has the smallest energy consumption. Hence, it is most efficient in this regard. Meanwhile, WOA consumes a little more energy than QSAS. Further, ACO has the moderate energy requirement. On the other hand, VM has the highest energy consumption, meaning that it is least energy efficient among the other compared algorithms. This implies that QSAS is the best choice whenever energy efficiency is a concern and VM is the most undesired choice.

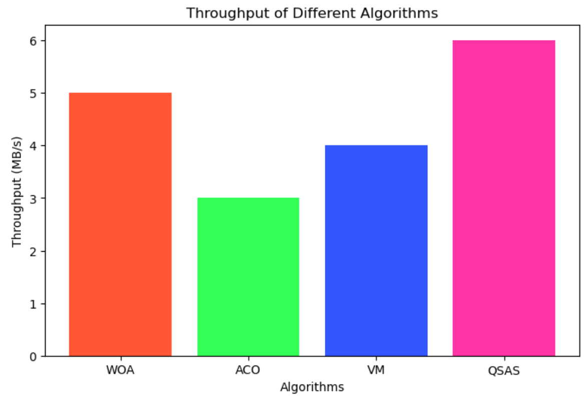

Figure 7: Throughput comparison chart

Figure 7 shows the throughput in MB/s for four algorithms namely WOA, ACO, VM, and QSAS. Throughput is the rate at which the data is processed; the higher the value, the better is the throughput. The presented QSAS method has the highest throughput at approximately 6 MB/s. Therefore, in terms of the rate of processing a given amount of data, QSAS is the most efficient algorithm. Further, WOA is a bit closer with almost 5 MB/s but VM has about a 4-MB/s throughput. ACO has been worst in terms of throughput, going only up to around 3 MB/s. So, QSAS outstands with the best performance in high-speed data processing, and ACO signifies at the very end with low throughput.

V CONCLUSION

This paper demonstrates that the comparison of different cloud resource management algorithms proves the QSAS outperformance over the traditional methods like WOA, ACO, and VM. Quantum State-based Adaptive Scheduling outperformance has the smallest latency of 30 ms, which ensures the shortest response time of the cloud tasks, and the lowest storage overhead of 5%, which reduces extra data storage expenses and increases the efficiency of the system. Also, QSAS presents the best energy consumption values, with 0.02 J/MB, which is an energy-efficient solution for large-scale cloud infrastructures. More importantly, it achieves the highest throughput of 6 MB/s and significantly improves the capability of data processing.

In comparison with other algorithms, WOA performs moderately well but lags behind QSAS on all the performance attributes. ACO has the highest latency of 70 ms and energy consumption of 0.1 J/MB, making it less efficient. VM has the worst values for all considered metrics: latency is 100 ms, storage overhead is 20%, and energy consumption is 0.15 J/MB. Such a comparison will highlight that QSAS indeed brings the adaptiveness and efficiency in the scheduling of cloud resources by its quantum-inspired decision-making mechanism that optimizes performance dynamically.

The results affirm that QSAS is the most effective solution for modern cloud environments where resource optimization, scalability, and real-time adaptability are critical. Combining the principles of quantum, QSAS further provides more flexibility, reduces system delays, and assures that the operations of the cloud are cost-efficient. This new approach in the area of cloud computing is revolutionary and opens ways for next-generation resource management techniques

REFERENCES

1. Madni, S. H. H., Abd Latiff, M. S., & Coulibaly, Y. (2016). Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. *Journal of Network and Computer Applications*, *68*, 173-200.
2. Mubeen, A., Ibrahim, M., Bibi, N., Baz, M., Hamam, H., & Cheikhrouhou, O. (2021). Alts: An adaptive load balanced task scheduling approach for cloud computing. *Processes*, *9*(9), 1514.
3. Mahmood, A., Khan, S. A., & Bahlool, R. A. (2017). Hard real-time task scheduling in cloud computing using an adaptive genetic algorithm. *Computers*, *6*(2), 15.
4. Liu, S., & Wang, N. (2020). Collaborative optimization scheduling of cloud service resources based on improved genetic algorithm. *IEEE Access*, *8*, 150878-150890.
5. Duan, K., Fong, S., Siu, S. W., Song, W., & Guan, S. S. U. (2018). Adaptive incremental genetic algorithm for task scheduling in cloud environments. *Symmetry*, *10*(5), 168.
6. Madni, S. H. H., Latiff, M. S. A., Ali, J., & Abdulhamid, S. I. M. (2019). Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. *Arabian Journal for Science and Engineering*, *44*, 3585-3602.
7. Zhu, J., Li, Q., Ying, S., & Zheng, Z. (2024). Research on Parallel Task Scheduling Algorithm of SaaS Platform Based on Dynamic Adaptive Particle Swarm Optimization in Cloud Service Environment. *International Journal of Computational Intelligence Systems*, *17*(1), 260.
8. Mangalampalli, S., Karri, G. R., & Kumar, M. (2023). Multi objective task scheduling algorithm in cloud computing using grey wolf optimization. *Cluster Computing*, *26*(6), 3803-3822.
9. Jia, L., Li, K., & Shi, X. (2021). Cloud computing task scheduling model based on improved whale optimization algorithm. *Wireless Communications and Mobile Computing*, *2021*(1), 4888154.
10. Mukherjee, D., Ghosh, S., Pal, S., Aly, A. A., & Le, D. N. (2022). Adaptive scheduling algorithm based task loading in cloud data centers. *IEEE Access*, *10*, 49412-49421.
11. Xu, F., Yin, Z., Gu, A., Li, Y., Yu, H., & Zhang, F. (2021). Adaptive scheduling strategy of fog computing tasks with different priority for intelligent production lines. *Procedia Computer Science*, *183*, 311-317.
12. Chen, Z., Lin, K., Lin, B., Chen, X., Zheng, X., & Rong, C. (2020). Adaptive resource allocation and consolidation for scientific workflow scheduling in multi-cloud environments. *IEEE Access*, *8*, 190173-190183.
13. Mourtzis, D. (2020). Adaptive scheduling in the era of cloud manufacturing. *Scheduling in industry 4.0 and cloud manufacturing*, 61-85.
14. Kruekaew, B., & Kimpan, W. (2022). Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning. *IEEE Access*, *10*, 17803-17818.
15. Zhu, L., Huang, K., Hu, Y., & Tai, X. (2021). A self-adapting task scheduling algorithm for container cloud using learning automata. *IEEE Access*, *9*, 81236-81252.
16. Li, W., Fan, Q., Dang, F., Jiang, Y., Wang, H., Li, S., & Zhang, X. (2022). Multi-Objective Optimization of a Task-Scheduling Algorithm for a Secure Cloud. *Information*, *13*(2), 92.
17. Aggarwal, A., Dimri, P., Agarwal, A., & Bhatt, A. (2021). Self adaptive fruit fly algorithm for multiple workflow scheduling in

cloud computing environment. *Kybernetes*, *50*(6), 1704-1730.

18. Abdullahi, M., Ngadi, M. A., Dishing, S. I., & Abdulhamid, S. I. M. (2023). An adaptive symbiotic organisms search for constrained task scheduling in cloud computing. *Journal of ambient intelligence and humanized computing*, *14*(7), 8839-8850.
19. Amer, A. A., Talkhan, I. E., Ahmed, R., & Ismail, T. (2022). An optimized collaborative scheduling algorithm for prioritized tasks with shared resources in mobile-edge and cloud computing systems. *Mobile Networks and Applications*, *27*(4), 1444-1460.
20. Hao, Y., Wang, L., & Zheng, M. (2016). An adaptive algorithm for scheduling parallel jobs in meteorological Cloud. *Knowledge-Based Systems*, *98*, 226-240.
21. Chen, L., Guo, K., Fan, G., Wang, C., & Song, S. (2020). Resource constrained profit optimization method for task scheduling in edge cloud. *IEEE Access*, *8*, 118638-118652.
22. Chen, X., Cheng, L., Liu, C., Liu, Q., Liu, J., Mao, Y., & Murphy, J. (2020). A WOA-based optimization approach for task scheduling in cloud computing systems. *IEEE Systems journal*, *14*(3), 3117-3128.
23. Dewangan, B. K., Agarwal, A., Choudhury, T., & Pasricha, A. (2020). Cloud resource optimization system based on time and cost. *International Journal of Mathematical, Engineering and Management Sciences*, *5*(4), 758-768.
24. Garg, V., & Jindal, B. (2023). Resource optimization using predictive virtual machine consolidation approach in cloud environment. *Intelligent Decision Technologies*, *17*(2), 471-484.
25. Junaid, M., Sohail, A., Rais, R. N. B., Ahmed, A., Khalid, O., Khan, I. A., ... & Ejaz, N. (2020). Modeling an optimized approach for load balancing in cloud. *IEEE access*, *8*, 173208-173226.
26. Kashyap, S., Singh, A., & Gill, S. S. (2025). Machine learning-centric prediction and decision based resource management in cloud computing environments. *Cluster Computing*, *28*(2), 130.
27. Lahande, P. V., Kaveri, P. R., Saini, J. R., Kotecha, K., & Alfarhood, S. (2023). Reinforcement Learning Approach for Optimizing Cloud Resource Utilization With Load Balancing. *IEEE Access*.
28. Lattuada, M., Barbierato, E., Gianniti, E., & Ardagna, D. (2020). Optimal resource allocation of cloud-based spark applications. *IEEE Transactions on Cloud Computing*, *10*(2), 1301-1316.
29. Prasad, V. K., Dansana, D., Bhavsar, M. D., Acharya, B., Gerogiannis, V. C., & Kanavos, A. (2023). Efficient Resource Utilization in IoT and Cloud Computing. *Information*, *14*(11), 619.
30. Qiu, M., Kung, S. Y., & Gai, K. (2020). Intelligent security and optimization in Edge/Fog Computing. *Future generation computer systems*, *107*, 1140-1142.
31. Ramamoorthi, V. (2021). AI-Driven Cloud Resource Optimization Framework for Real-Time Allocation. *Journal of Advanced Computing Systems*, *1*(1), 8-15.
32. Reddy, P. V., & Reddy, K. G. (2023). A multi-objective based scheduling framework for effective resource utilization in cloud computing. *IEEE Access*, *11*, 37178-37193.
33. Sangaiah, A. K., Hosseinabadi, A. A. R., Shareh, M. B., Bozorgi Rad, S. Y., Zolfagharian, A., & Chilamkurti, N. (2020). IoT resource allocation and optimization based on heuristic algorithm. *Sensors*, *20*(2), 539.
34. Sharma, N., & Garg, P. (2022). Ant colony based optimization model for QoS-Based task scheduling in cloud computing environment. *Measurement: Sensors*, *24*, 100531.
35. Shukur, H., Zeebaree, S., Zebari, R., Zeebaree, D., Ahmed, O., & Salih, A. (2020). Cloud computing virtualization of resources allocation for distributed systems. *Journal of Applied Science and Technology Trends*, *1*(2), 98-105.
36. Suryavanshi, R., & Chawla, P. (2025). Optimizing Cloud Networking for Large Language Models: The Role of AI-Driven Solutions. *Asian American Research Letters Journal*, *2*(1), 1-7.

37. Ugbebor, F. O. (2024). Intelligent Cloud Solutions Bridging Technology Gaps for Small and Medium-Sized Enterprises. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *7*(01), 161-186.
38. Xiong, X., & Yang, G. (2024). A node deployment and resource optimization method for CPDS based on cloud-fog-edge collaboration. *IET Generation, Transmission & Distribution*, *18*(21), 3524-3537.