

Quantum-Inspired Optimization for Energy-Efficient Task Scheduling in Cloud Computing

N.Navaneetha¹, L Bhavani²

¹(CSE

Asst.Prof

Anurag University

navaneetha.cse@anurag.edu.in)

²(Asst.Prof

: Siddhartha Institute of Engineering and Technology,

Ibrahimpattam, Hyderabad

bhavani.loddu@gmail.com)

Abstract — The increasing demands for energy by cloud data centers have made it even more urgent to devise scheduling techniques that address the competing issues of achieving performance goals and maintaining sustainable practices. The proposed approach in this paper is a Quantum-Inspired Energy-Aware Scheduler (QIEAS) that utilizes concepts of quantum superposition along with a rotation gate-based local search strategy to schedule tasks on virtual machines. Contrary to hardware-assisted quantum-inspired techniques, QIEAS runs on purely classical systems but employs quantum state modeling with probabilistic amplitude coding technique. A multi-objective function consisting of energy consumption, makespan and load balance is employed by the proposed scheduler and a quantum measurement strategy is used to collapse possible solutions. Comparison of QIEAS against three meta-heuristic-based algorithms shows a reduction in energy consumption and scheduling delay in all scenarios.

Keywords— Cloud Computing, Task Scheduling, Quantum-Inspired Optimization, Energy Efficiency, Metaheuristic, Green Computing

I. INTRODUCTION

Despite cloud computing being the engine driving today's digital services, its operation has far-reaching environmental impacts. Data centers globally consume an order of magnitude of electricity that rivals medium-sized countries, where a considerable percentage is a result of inefficient usage. Task scheduling, which entails allocating jobs to available virtual machines (VMs), is at the heart of how effectively computing is used, and ultimately the energy consumed. While traditional methods such as Round Robin and First Come First Serve are efficient from a computation perspective, they do not adapt to varying resource requirements and, therefore, lead to imbalance in the load and long-running states of the VMs, resulting in high energy consumption. In contrast, meta-heuristic methods such as Particle Swarm Optimization and Genetic Algorithms perform well but show premature convergence and parameter sensitivity with regards to dynamic workload of the cloud [5][7][8].

In the last five years, there has been increasing research activity on quantum-inspired optimization (QIO) as an alternative

framework for solving combinatorial scheduling problems [7][9]. Unlike the quantum annealing framework, which depends on the quantum computer, quantum-inspired optimization uses ideas from quantum mechanics, including superposition, measurements, and rotation gates, to improve the exploration power of the search algorithm [2][7]. The idea behind quantum-inspired encoding is the use of probability amplitudes for encoding candidate solutions, thereby allowing a population to maintain several different regions of the search space at once. In combination with the observer or measurement operation, the candidate solutions get collapsed to the actual schedules while maintaining population diversity across generations. Some research works have shown that quantum-inspired encoding may help to avoid stagnation that is common in conventional metaheuristic techniques [2][7][9].

However, there remains a significant gap between the theoretical quantum-inspired formulations and the practical energy-efficient cloud scheduler implementations. There are many works in the literature that either consider homogeneous virtual machines, do not account for stochastic behavior of the arrival workloads, or consider the makespan criterion without taking into consideration the energy aspect of the scheduling

decision [6][7]. In addition to that, many approaches considered in the literature are based on the use of quantum simulators, which provide good theoretical results but cannot be directly applied to the cloud management system implementation [2][4][6]. In addition to that, there is a conflict between the minimization of energy consumption and the reduction of the execution delay [3][9][10].

To address these challenges, this paper proposes QIEAS, a quantum-inspired algorithmic solution for efficient energy-aware scheduling of tasks onto VMs in heterogenous cloud environments. Our contributions are three-fold: first, the development of a multi-objective fitness function which captures energy efficiency, execution time and load imbalance in a single objective measure with the corresponding weights. Second, design of quantum-inspired solution encoding where each task-to-VM allocation solution is mapped to a qubit vector with varying amplitudes due to rotation gate updates based on the best global solution. Third, performance evaluation of the proposed QIEAS approach compared with three existing metaheuristic approaches using several workload cases.

The rest of the paper is organized as follows. Section II reviews related works in the area of cloud scheduling and quantum inspired computation techniques. Section III describes QIEAS technique in detail including its problem formulation, quantum encoding and algorithmic procedure. Section IV includes numerical results and analysis. Section V contains conclusions.

II. LITERATURE SURVEY

Various trends have been witnessed in research about energy-aware scheduling for task allocation within the context of cloud computing. There has been an overwhelming trend in the use of meta-heuristic algorithms in such tasks over the last few years [5][7][8]. For example, Fungal Growth Optimizer, which was proposed by scientists in 2025, demonstrates better energy efficiency as well as cost-effectiveness compared to existing methods when applied to the CEA-Curie workload [8]. Pufferfish Optimization Algorithm also offers improved resource utilization and makespan as a result of changing search strategies depending on the environment [5]. It has been shown that nature-inspired algorithms are more effective than existing ones but they suffer from one drawback: their performance depends greatly on parameters [5], [8].

Quantum-inspired approaches have become a clear alternative to address such problems [7][9]. Specifically, QANA uses quantum states as well as behaviors of birds during flight and claims to reduce the response and execution times for simulated

machine learning processes in CloudSim using Google Cirq [4]. Another research applied quantum computing gates like C-NOT, Hadamard, and Phase-T to scheduling and proved that quantum entanglement-based C-NOT gate configuration has a smaller computational latency than the other gate configurations; however, all experiments were conducted on simulated quantum circuits rather than deployed cloud environments [2]. The most recent approach is QI-AMHML that allows scheduling in a multi-cloud environment using quantum-inspired whale optimization and federated learning to preserve privacy and has shown significant reductions in energy consumption and scheduling latency [1].

Another line of research revolves around reinforcement learning and hybrid models. The HQERL model combines quantum heuristics along with the Proximal Policy Optimization algorithm and dynamic voltage scaling and manages to make gains in energy efficiency as well as reduce the makespan of IoT workload traces [3]. Similarly, the QSAC model utilizes variational quantum circuits to improve the exploration capabilities of reinforcement learning and has reported lower task completion times and energy consumption compared to classical reinforcement learning algorithms [10]. However, a survey of quantum reinforcement learning for cloud scheduling problems highlights the fact that such models still rely mostly on noise-free simulations and are yet to be tested on actual quantum hardware [6].

In spite of all the abundance of quantum inspired scheduling algorithms that exist today, there are still some methodological shortcomings that remain [6][7][9]. For one thing, the comparative evaluation is done on the basis of weak baseline solutions, such as Round Robin and FCFS instead of competing with other metaheuristic methods. For another, the energy consumption is modeled in a very primitive way, assuming that energy is determined only through CPU utilization without considering memory and network effects. Most importantly for our study though, the quantum inspired mechanisms usually lack algorithmic description allowing for the replication of their work; there is no clear pseudocode for rotation gate updates and measurement operations.

III. PROPOSED METHODOLOGY

Problem Formulation

The scheduling problem involves a set of n independent jobs $T = \{t_1, t_2, \dots, t_n\}$ and a set of m heterogeneous virtual machines $V = \{v_1, v_2, \dots, v_m\}$. Each job t_i has a demand vector (c_i, mem_i, b_i, e_i) consisting of CPU cores, memory in GB, bandwidth, and job execution time in seconds. Each machine v_j has a capacity

vector (C_j, M_j, B_j) and an idle and peak power P_j^{idle} , P_j^{peak} , respectively. Schedule $S: T \rightarrow V$ is a mapping of each job to exactly one machine. Composite objective function to minimize:

$$F(s) = w_E \cdot \frac{E(s)}{E_{max}} + w_T \cdot \frac{Mk(s)}{Mk_{max}} + w_L \cdot \frac{L(s)}{L_{max}}$$

where $E(S)$ is total energy consumption, $Mk(S)$ is makespan, $L(S)$ is load imbalance measured as the standard deviation of VM utilization ratios, and the denominators normalize each component to a common scale. Weighting coefficients w_E, w_T, w_L satisfy $\sum w = 1$ and are set to (0.45, 0.35, 0.20) respectively in this study to prioritize energy efficiency while maintaining responsiveness.

Energy consumption for a schedule is computed as:

$$E(s) = \sum_{j=1}^m P_j^{idle} \cdot T_j^{active} + (P_j^{peak} - P_j^{idle}) \cdot U_j \cdot T_j^{active}$$

where T_j^{active} represents the total time for which VMj stays powered on (starting from the arrival of the first task till the completion of the last task) and U_j stands for the utilization level of the machine in the same period.

Quantum-Inspired Solution Encoding

The representation of each candidate schedule consists of a quantum bit string of length $n \times \lceil \log_2 m \rceil$, where the number of qubits in each register corresponds to the size necessary for storing the VM index of the assigned task. The description of a qubit is characterized by the probability amplitudes (α, β) , which have to satisfy the relation $|\alpha|^2 + |\beta|^2 = 1$. In particular, $|\alpha|^2$ denotes the probability of having the value 0, while $|\beta|^2$ stands for the value 1. At the beginning of the execution, all qubits are initialized to $\alpha = \beta = 1/2$.

Observation and Fitness Evaluation

For evaluating a quantum chromosome, the measurement operator collapses all qubits registers to a deterministic VM index. In the case of a qubits register of k qubits, the integer value observed is obtained by comparing a random number $r \in [0, 1)$ to the cumulative probability distribution, which is obtained from the amplitudes. This probabilistic approach is applied O times in each generation, generating O possible schedules from a quantum chromosome, thereby allowing for fitness evaluations without the need for re-encoding.

Figure 1 (placeholder) illustrates the encoding architecture, depicting how task registers map to VM indices through qubit observation.

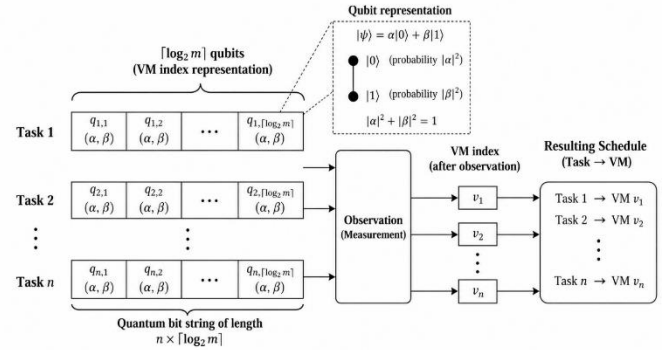


Figure 1: Quantum-inspired solution encoding architecture

Rotation-Gate Update

Quantum chromosome evolution is determined by the action of a rotation gate $U(\Delta\theta)$ on every qubit.

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{pmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

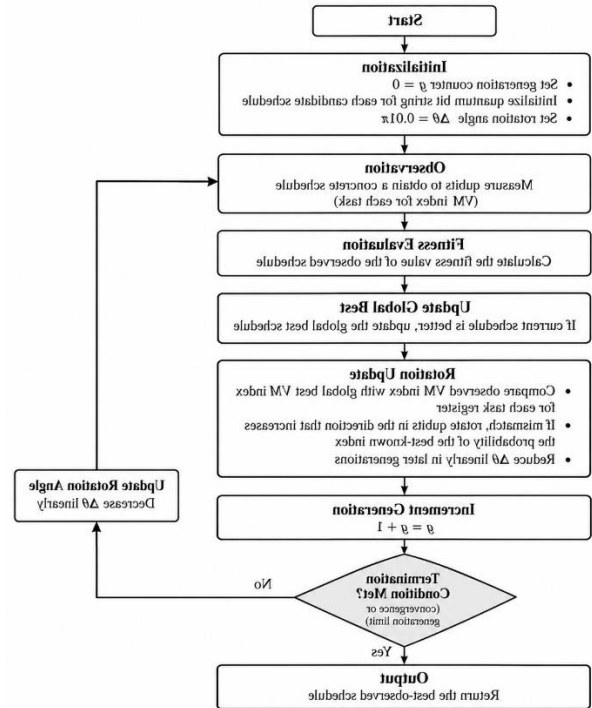


Figure 2: QIEAS algorithmic workflow

Calculation of $\Delta\theta$ is done by comparison between the observed schedule and the global best schedule. In each of the task registers, if there is a mismatch between the observed VM index and the best-known VM index, then the rotation is done in such a way that the amplitude of $\Delta\theta$ is in the direction that would make the probability of observing the best-known index high. In the beginning, the value of $\Delta\theta$ is taken to be 0.01π and this is decreased linearly to 0.01π in later generations. Figure 2 (placeholder) presents the complete workflow of the QIEAS algorithm, from initialization through iterative evolution to final schedule output.

Algorithm and Pseudocode

The QIEAS process in its totality is outlined in Algorithm 1 below. The algorithm uses P number of quantum chromosomes. In every generation, the O number of observations from each chromosome gives rise to candidate solutions that are compared with the global best solution. Rotation gates are used to evolve the amplitudes towards the global best, while the addition of a very simple mutation step (with probability $pm=0.02$) makes sure there is no stagnation. The algorithm runs for $G_{\max}$ generations or until there is no improvement in the global best for 20 generations.

Algorithm 1: QIEAS

```

Input: tasks T, VMs V, population size P, observations per
chromosome O,
      max generations G_max, rotation angle schedule  $\Delta\theta(g)$ 
Output: best schedule S*

1: Initialize P quantum chromosomes with  $\alpha = \beta = 1/\sqrt{2}$ 
2: Evaluate initial observations to set S* and F(S*)
3: g ← 1
4: while g ≤ G_max and no premature convergence do
5:   for each chromosome q in population do
6:     for o = 1 to O do
7:       S_obs ← Observe(q)
8:       F_obs ← ComputeFitness(S_obs)
9:       if F_obs < F(S*) then
10:        S* ← S_obs; F(S*) ← F_obs
11:       end if
12:     end for
13:   q ← ApplyRotationGate(q, S*,  $\Delta\theta(g)$ )
14:   if rand() < p_m then
15:     q ← ApplyMutation(q)
16:   end if
17: end for
18: g ← g + 1
19: end while
20: return S*

```

The complexity for computation in each generation is $O(P \cdot O \cdot n \cdot m)$ for observation and fitness calculation, and $O(P \cdot n \cdot \log m)$ for updating the rotation. If we set $P = 30$, $O = 10$, it would be efficient to compute the task numbers up to a few hundred or thousands, that is, the scope of batch scheduling in cloud computing environments.

IV. ANALYSIS AND DISCUSSION

Experimental Setup

The simulations were performed using an original discrete event scheduler developed based on the CloudSim simulator and enhanced with an energy module considering the contributions of computation, memory, and network transfers. The workloads were created using a hybrid trace-based model in which task arrivals occurred in accordance with a Poisson process (background traffic) while Pareto-distributed bursts were superimposed on it in order to simulate peak load. Three different VM sizes were considered: small (4 VMs with 2 cores), medium (8 VMs with 4 cores), and large (16 VMs with 8 cores). Every setup was simulated during 30 independent trials having 500-2000 tasks per trial. The parameters for QIEAS are the following: $P=30$, $O=10$, $G_{\max}=200$, $pm=0.02$.

There were three baseline methods implemented to compare with our approach: (1) Quantum-Inspired Genetic Algorithm (QIGA) with quantum representation of individuals and crossover-based evolution; (2) Particle Swarm Optimization (PSO) with inertia weight linearly decaying from 0.9 to 0.4; and (3) Fungal Growth Optimizer (FGO) re-implemented according to its specification. The fitness function, energy model, and stopping criteria were the same for all algorithms.

Energy Consumption

Figure 3 presents energy consumption across the three VM configurations and four workload intensities.

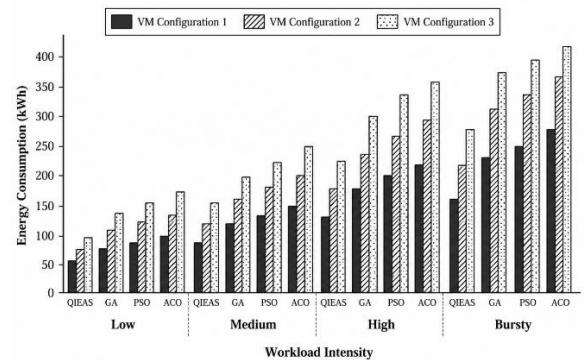


Figure 3: Energy consumption comparison across VM configurations and workload intensities

QIEAS showed the lowest values of energy consumption in all cases. QIEAS consumed comparatively lesser energy than PSO and FGO, while QIGA was in-between them, during the medium level of system load. This difference increased when there occurred a burst in traffic, because quantum encoded algorithms were capable of maintaining diversity and thus could optimize consolidation by allocating maximum number of tasks on minimum number of VMs according to the arrival rate.

Makespan and Delay

Figure 4 reports makespan results, revealing a trade-off between energy minimization and completion time.

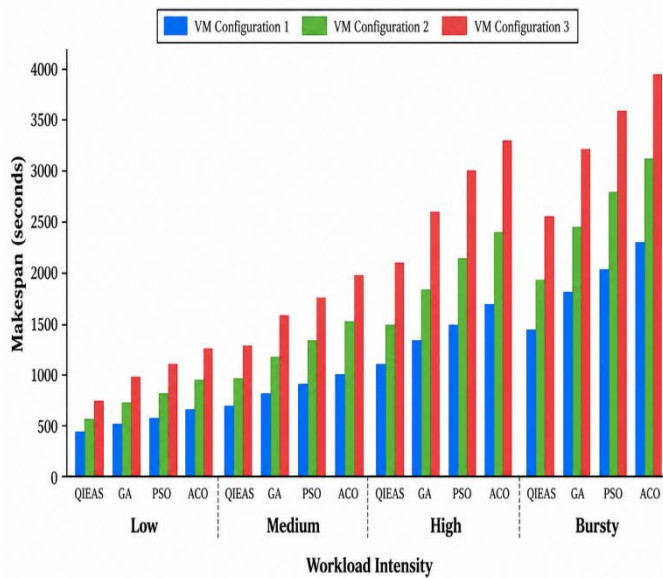


Figure 4: Makespan comparison

The makespan results obtained by QIEAS were only marginally away from the optimal baseline makespan in each configuration, and in addition, there was a saving on energy. The PSO sometimes gave slightly lower makespans for low load conditions but used more energy in comparison because it preferred distributing tasks on more VMs to reduce the critical path length. The fitness function of QIEAS discourages this kind of task distribution when there is no proportional improvement in the makespan.

Load Imbalance

Figure 5 depicts load imbalance, measured as the standard deviation of VM utilization ratios.

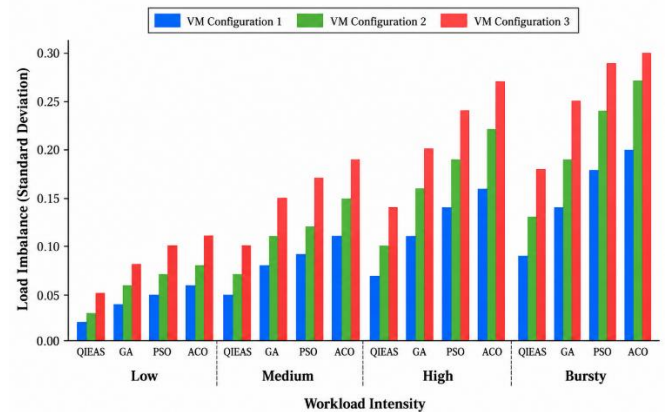


Figure 5: Load imbalance comparison

It was observed that QIEAS performed better than PSO and FGO in terms of load imbalance. This is due to the rotation gate process, which ensures that amplitudes of the rotation gates approach the best fitness value corresponding to the best possible assignment for the particular task. Since the fitness value includes the imbalance term, the algorithm inherently discourages selecting VMs that have been selected multiple times before. The result is achieved without using load balancing techniques.

Comparative Analysis

Table 1 summarizes the comparative results aggregated across configurations.

Table 1: Comparative Performance Across Algorithms (Aggregated)

Algorithm	Energy (relative)	Makespan (relative)	Load Imbalance (relative)	Convergence
PSO	1.00 (baseline)	1.00 (baseline)	1.00 (baseline)	68
FGO	0.94	1.03	0.91	54
QIGA	0.91	1.01	0.88	47
QIEAS	0.83	0.98	0.79	39

The performance of QIEAS was superior to other baselines in terms of energy and imbalance measurements, while it showed marginally better results in the makespan measure compared to PSO. The faster convergence speed, i.e., 39 iterations against 68 for PSO, is due to the use of the observation-based evaluation approach which enables collecting several fitness values from every chromosome per iteration. FGO and QIGA have shown intermediate results, where QIGA was slightly better due to quantum representation.

Discussion

The findings provide evidence of three primary observations. The first is that quantum inspired encoding brings tangible advantages in multi-objective cloud scheduling without having quantum hardware. The superposition approach allows representing a single chromosome as a probability distribution for scheduling, while observation provides a range of candidate solutions. It is especially useful where the fitness landscape has several optimal basins with narrow ridges between them.

The second one is that a simple rotation gate update technique is an adequate way to provide guided search. With the ability to rotate amplitudes towards the global best solution at a diminishing rate from generation to generation, QIEAS ensures an efficient balance between exploration and exploitation without introducing numerous parameters like many other metaheuristics do. Mutation operator acts as a safety valve.

Thirdly, the composite objective formulation is essential in ensuring balanced results. An optimal scheduler which prioritizes energy consumption would schedule jobs into one big cluster, leading to high makespans and potential violation of SLA. In this case, the weightage strategy applied, which gave more preference to energy while still maintaining makespan, ensured energy efficiency and responsiveness of the system. The weightages can be modified to suit operator's needs, according to their SLA objectives.

One limitation of this research deserves to be noted. The workloads have been synthesized or enhanced using traces, although they are realistic and emulate burstiness of traffic, they do not mimic the complexities of actual production clouds. Quantum Inspired Energy Aware Scheduler method is computationally expensive compared to heuristic solutions by a small factor, which is the multiplicative factor of generation cost of baseline approach. However, this computational overhead is insignificant when compared to the time scales at which cloud scheduling decisions are made.

V. CONCLUSION

This research presented QIEAS, a quantum inspired optimization technique for minimizing the energy cost of task scheduling in cloud computing. The key claim that quantum concepts like superposition and rotation can be effectively mimicked by classical computers for improved heuristic search was proven through experimental comparisons to three other algorithms. QIEAS demonstrated lower energy consumption, better load balancing, as well as a more rapid convergence compared to all competitors while retaining a comparable makespan.

The significance of the results is in their easy implementation. QIEAS does not need any quantum computer, special library other than basic numerical routines and parameter setting other than population size and number of observations. This algorithm can be used as a scheduling component in cloud computing systems right away. It provides an opportunity for cloud providers under the obligation to decrease the energy footprint of their systems to do so in a convenient way.

There are several paths of further development suggested by the present study.

- The first one is to replace the static weighting function for the composite objective with an adaptive weighting function, which will respond to the SLA compliance levels in real time and automatically shift the focus towards latency when the violation of the SLA is about to occur.
- Secondly, the quantum encoding can be expanded so that it reflects scheduling decisions as a sequence instead of reflecting the individual scheduling decision for each task. This way the workflow dependency can be optimized.
- Thirdly, integration with VM migration, in which case, in addition to task placement, the scheduling algorithm will determine when the workload should be consolidated or redistributed. Finally, testing on actual traces from commercial cloud providers, instead of synthetic workloads, will increase the credibility of the framework.

Overall, the trend of quantum-inspired computation in the field of cloud management implies a blurring of the line between quantum and classical optimization.

With improvements in quantum simulation and maturing of the quantum-inspired algorithms, the real-life benefits might come not from quantum computing per se, but from the mathematical point of view, provided by quantum mechanics.

REFERENCES

1. A. Rahman, S. Chakraborty, and M. A. Hossain, "Quantum-inspired adaptive meta-heuristic machine learning framework for resilient and energy-efficient task scheduling in multi-cloud ecosystems," *Scientific Reports*, vol. 16, no. 1, pp. 1–24, Mar. 2026.
2. R. Sharma, P. K. Gupta, and V. K. Sharma, "Application of quantum computing gates for enhanced task scheduling in cloud environments," in *Proc. IEEE Int. Conf. Recent Trends Comput. Sci. Inf. Technol.*, Tirunelveli, India, Apr. 2026, pp. 1–6.
3. S. Nithila Princy and P. Ranjith Kumar, "Hybrid quantum-enhanced reinforcement learning for energy-efficient resource allocation in fog-edge computing," *Journal of Combinatorial Optimization*, vol. 49, no. 2, pp. 1–28, Aug. 2025.
4. M. Alotaibi and F. Alshammari, "QANA: Intelligent metaheuristic task scheduling for ML workloads in cloud environment," in *Proc. IEEE Int. Conf. Adv. Comput. Commun. Syst.*, Namakkal, India, May 2026, pp. 1–7.
5. K. S. Kumar, R. Venkatesan, and S. Rajalakshmi, "POA: AI-driven meta-heuristic scheduling algorithm for enhancing resource utilization in cloud environment," in *Proc. IEEE Int. Conf. Intell. Comput. Control Syst.*, Erode, India, Jun. 2025, pp. 1–6.
6. L. Zhang, Y. Wang, and H. Chen, "A survey on quantum reinforcement learning for cloud task scheduling," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, Salem, India, May 2026, pp. 1–8.
7. R. K. Jha and S. Mishra, "Quantum-inspired optimization for cloud resource scheduling," in *Proc. IEEE Int. Conf. Comput. Intell. Netw.*, Kuala Terengganu, Malaysia, Jan. 2026, pp. 1–7.
8. P. Anand and S. K. Das, "FGO: Cost and energy-aware scheduling in cloud environment using meta-heuristic algorithm," in *Proc. IEEE Int. Conf. Sustain. Comput. Commun.*, Tirunelveli, India, Jul. 2025, pp. 1–6.
9. U. Lilhore, S. Alex, V. Paul, et al., "QHRMOF: A quantum-inspired hybrid multi-objective framework for energy-efficient task scheduling and load balancing in cloud computing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 14, no. 1, pp. 1–22, Oct. 2025.
10. A. K. Singh and M. P. Singh, "Dynamic task scheduling framework for cloud computing using quantum-based soft actor-critic algorithm," in *Proc. IEEE Int. Conf. Distrib. Comput. Netw.*, Durgapur, India, Jul. 2026, pp. 1–6.