Proposing a Meta-heuristic Algorithm Focused on Energy Consumption Improvement in Cloud Resource Scheduling

Rahmat Zolfaghari
Department of Computer Engineering, Has.C.,
Islamic Azad University, Hashtgerd, Iran

Hamed Badershah
Department of Computer Engineering, SR.C.
Islamic Azad University, Tehran, Iran

ABSTRACT
Cloud computing, as one of the most advanced computational technologies, provides extensive capabilities for resource sharing and scalability, but high energy consumption in cloud data centers has become one of the primary challenges. The goal of this research is to present a new meta-heuristic algorithm to optimize energy consumption and enhance resource efficiency in cloud environments. The proposed algorithm was implemented using simulations in real cloud environments, such as Amazon EC2 and Planet Lab. In this process, the proposed algorithm was compared with traditional algorithms like PSO, and three main metrics, including energy consumption, execution time, and resource efficiency, were evaluated. Simulation results showed that the proposed algorithm was able to reduce energy consumption by 15.8%, decrease task execution time by 14.6%, and increase resource efficiency by 10.8%.

Keywords
Cloud computing, Energy optimization, Meta-heuristic algorithm, Resource efficiency

1. INTRODUCTION AND RELATED WORKS
Cloud computing, as a transformative innovation in information technology, enables users to easily and scalably access computational resources [1]. With its unique advantages, such as cost reduction and enhanced efficiency, cloud computing has garnered significant attention in recent years [2]. However, the rapid expansion of data centers and cloud computing services has led to a substantial increase in energy consumption, which not only raises operational costs but also causes significant environmental impacts [3]. Consequently, optimizing energy consumption in cloud computing has become a critical challenge [4]. One effective approach to addressing this issue is optimizing the resource scheduling process [5], where computational tasks are allocated to available resources in a manner that enhances system efficiency and reduces energy usage [6]. In studies [7, 8] energy consumption in cloud system are studied. In [9] a Meta-heuristic method for energy consumption efficient in in cloud system is provided. In [10] a prediction method for predict and decreased energy consumption in cloud system is provided. Meta-heuristic methods, known for their high capacity to explore solution spaces and generate near-optimal solutions, have emerged as powerful tools for tackling complex, multi-objective problems such as resource scheduling in cloud environments. Algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) combine the strengths of local and global search to identify optimal solutions that minimize energy consumption while maintaining performance and service quality [11]. This paper aims to assess the performance of existing algorithms in resource scheduling and introduce a novel meta-heuristic algorithm that addresses specific limitations of current methods. Furthermore, the study explores the challenges associated with energy-efficient scheduling and proposes potential solutions to overcome these challenges. Optimizing cloud computing systems is essential in chat bots [12] and large-scale support systems [13] that have special applications in social networks.

2. RESEARCH METHODOLOGY
This research aims to explore and enhance cloud resource scheduling algorithms, with a specific focus on optimizing energy consumption. The cloud resource scheduling problem was initially modeled, and subsequently, the proposed algorithm was designed using meta-heuristic methods, particularly Particle Swarm Optimization (PSO). To assess the performance of the algorithms, precise simulations were carried out using CloudSim and MATLAB tools. Reliable data were gathered from well-established sources, such as Amazon EC2 and Planet Lab. Metrics including energy consumption, execution time, and resource utilization were used to evaluate the algorithms. Finally, the results from these simulations were analyzed using statistical techniques to compare the proposed algorithm's performance. The system model for proposed algorithm in Figure1 is portrayed. As Figure1, a numbers of user's request as VMs to be hosted and scheduler on PMs of cloud, optimally. As result, the energy consumption, response time, and resource utilization are optimized.

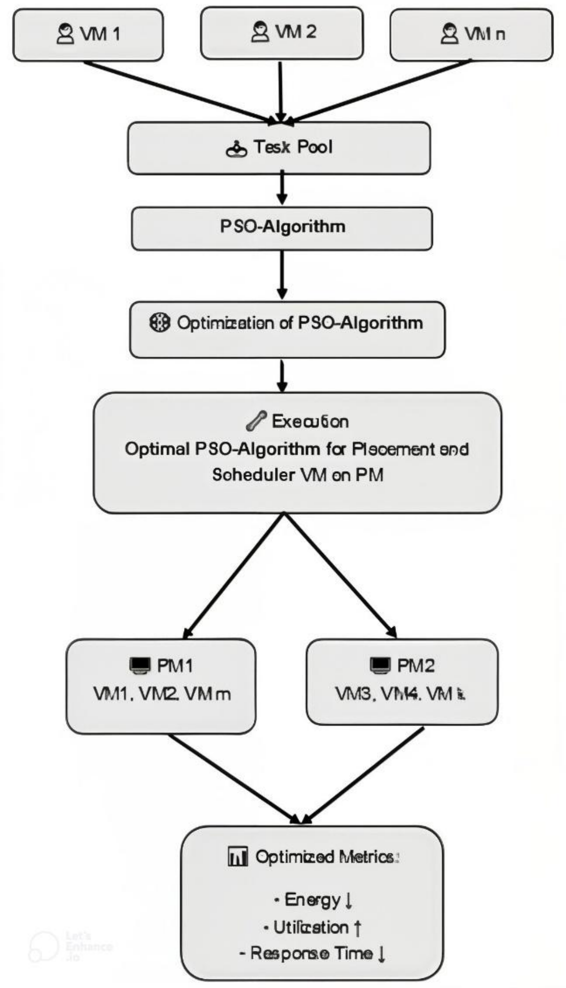

Figure 1. The system model of proposed algorithm

2.1 Problem Modeling and Simulation

During this phase, the key parameters that significantly influence the system's performance were simulated. These parameters included energy consumption, execution time, load balancing, resource accessibility, and Service Level Agreement (SLA). CloudSim and MATLAB were used in tandem for the simulation of the cloud environment. CloudSim was utilized for simulating various conditions and evaluating algorithm performance, while MATLAB was employed to develop and enhance the meta-heuristic algorithms. This dual approach provided a more comprehensive analysis of the algorithm's performance under real-world conditions.

2.2 Data Collection

Simulation data were collected from reliable sources such as Amazon EC2 and the Planet Lab Common Project. These datasets provided valuable information, including workload data, response times, and energy consumption values, which were essential for evaluating the performance of the proposed algorithm. The use of credible datasets ensured that the evaluations were in alignment with real-world conditions.

2.3 Development, Simulation, and Evaluation of the Proposed Algorithm

The proposed algorithm was designed as an energy-efficient meta-heuristic approach aimed at reducing energy consumption and enhancing resource utilization in cloud computing environments. Particle Swarm Optimization (PSO) was used to optimize resource allocation. The algorithm was then tested in a simulated environment using the collected data. During the testing process, key metrics such as energy consumption reduction, execution time improvement, and resource utilization were closely examined. The algorithm was simulated alongside existing methods to evaluate its performance. Simulations assessed the impact of the algorithms on critical metrics, including energy consumption, execution time, load balancing, and resource utilization. The simulation data were analyzed using MATLAB, and statistical techniques were applied to compare the performance of the proposed algorithm with that of existing solutions to evaluate its improvements and overall efficiency.

3. DESCRIPTION OF PROPOSED ALGORITHM

In this section, proposed algorithm Phases with detail is descripted.

3.1 Genetic Algorithm Framework and its' Implementation

In the genetic algorithm, chromosomes are composed of genes, and the population evolves by combining chromosomes. This optimization process uses three main operators: selection, crossover, and mutation. The selection operator identifies high-quality chromosomes based on their fitness function. During crossover, selected chromosomes combine their genes to create offspring that outperform their parents, driving the population toward a local optimum. Mutation introduces diversity by altering a single gene in a chromosome, preventing premature convergence. Mutation is performed in two ways: by modifying the task request matrix or by swapping frequently used genes with less frequently used genes in the virtual machine (VM) matrix, reducing data transfer delays. Once the new population is generated, chromosomes that exceed the deadline are discarded, and the remaining chromosomes are evaluated based on cost and execution time, with the best one selected. This process continues until the chromosomes converge to an optimal solution.

3.2 Simulation Setup and Matrix Definitions

Various matrices were utilized in the simulation. The number of requests was set to 96, with every 8 requests forming a chromosome, resulting in 12 chromosomes. The DAG matrix (10x10) represents task relationships and dependencies. The Data matrix (10x10) shows data transfer delays between tasks. The prk matrix (8x1) represents the processing power of virtual machines, while the w matrix (10x1) defines execution times for each task. The VM cost matrix (8x1) specifies the rental cost of each virtual machine. For simulation purposes, α and β are used to define the crossover and mutation rates in the genetic algorithm. Requests are leveled based on their dependencies, with tasks lacking outputs moved to the first level. The BOTtask matrix illustrates task arrangements from highest to lowest levels, and the update_VM matrix is designed to optimize execution time and minimize task completion times.

3.3 Implementation Structure of the Algorithm

The workflow for generating the initial population starts by introducing essential inputs such as matrices, α, β, iteration max, population size, prk parameters, and data. Tasks are ranked randomly based on prerequisites, and each task is placed in the appropriate level. Chromosomes are formed by combining ranked tasks, with the best chromosome selected based on cost and time constraints. Crossover and mutation operators are applied to optimize the population. If the new population is smaller than the initial one, crossover occurs with probability α, followed by mutation with probability β.

3.4 VM Task Assignment Policies

Four policies for assigning virtual machines (VMs) to tasks are considered. The first policy randomly selects the most expensive VM for each task, resulting in fast but costly scheduling. The second policy uses the least expensive VM for all tasks, reducing costs but increasing execution time. The third policy calculates the latest finish time (LFT) for each task and assigns VMs based on availability, starting with cheaper options and upgrading when necessary. The fourth policy uses a random VM from the most expensive type for all tasks, ensuring uniform VM types but leading to higher costs.

3.5 Task Creation and Ordering

A random "order task" is created by ranking requests based on dependencies, with tasks without outputs moved to the first level. This process continues until all tasks are ranked, and the best chromosomes are selected based on execution time and costs.

3.6 Calculation of Initial LFT

The LFT and execution times are calculated using matrices to compute transfer and execution times for each request. The LFT calculation starts from the last task and proceeds backward, ensuring that each task meets its time constraints while using appropriate virtual machines.

3.7 VM Pool Update and Resource Optimization

Once VMs are assigned, subsequent tasks are allocated VMs from the VM pool. If a VM cannot meet a task's deadline, a higher-performance VM is selected. The system ensures that tasks are completed on time while optimizing VM usage and minimizing costs.

4. SIMULATION OF THE PROPOSED ALGORITHM AND ITS' RESULTS

In this section, the proposed algorithm is simulated and simulation results as Figures 2 to 4 are shown.

4.1 PSO Algorithms and Evaluation Criteria

Cloud computing uses distributed computing, grid computing, and virtualization technologies, allowing efficient resource usage with minimal hardware. Challenges in cloud computing include scalability, load balancing, and performance. Task scheduling is critical for optimizing cloud resources, and Particle Swarm Optimization (PSO) is proposed for this purpose. The PSO algorithm uses particles with two key attributes: position (X) and velocity (V). The particles move toward the best solutions found by others, optimizing the search. The position and velocity of each particle are updated using the following equations as Formulas 1and 2.

$$V(t+1)W^{v}(t) + c1 * R1 * \left(pbest(t) - p(1)\right) + C2 * R2 * (gbest(t) = p(t)) \quad (1)$$

$$p(t+1) = p(t)v(t+1) \quad (2)$$

Where V represents particle velocity, W is the weight coefficient, P is the current position, pbest and gbest represent the best local and global solutions, respectively, R1 and R2 are random numbers within [0,1], and C1 and C2 are learning coefficients.

4.2 Simulation Environment, Scenarios, and Results

The CloudSim simulator, which supports both Windows and Linux, was used to simulate task scheduling algorithms. The performance of the proposed PSO method was compared with the standard PSO algorithm based on metrics such as total time, energy consumption, and resource usage. Simulation setups were adjusted with varying parameters, including the number of nodes. For the first scenario, the number of nodes was varied, with the configuration parameters is shown as Table 1.

Table 1: Simulation Parameters for the First Scenario.

Parameter	Values
Number of VMs	75
Number of Tasks	25, 50, 75, 100
Simulation Time	S900
MAC Layer	802.11
Queue Type	Drop Tail
Buffer Size	50 packets
Node Stop Time	0 seconds
Node Placement	Random

Simulation results, shown in Figures 2, 3, and 4, compare total time, resource utilization, and energy consumption between the PSO and the proposed method. The results highlight the superior performance of the proposed method, with improvements in time efficiency, resource utilization, and energy consumption reduction compared to the standard PSO algorithm. In Figures 2 comparison of total time between PSO and proposed method is shown; In Figures 3 comparison of resource utilization between PSO and proposed method is shown; In Figures 4 comparison of energy consumption between PSO and proposed method is shown. As these figures, the energy consumption, response time, and resource utilization in cloud are optimized.

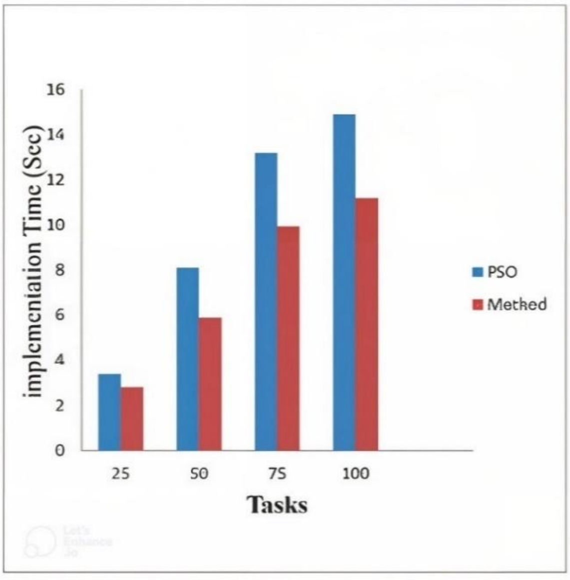

Figure 2: Comparison of total time between PSO and proposed method

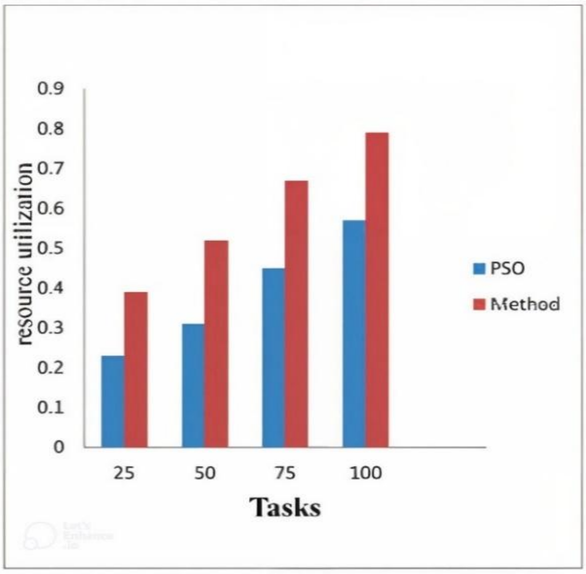

Figure 3: Comparison of resource utilization between PSO and proposed method

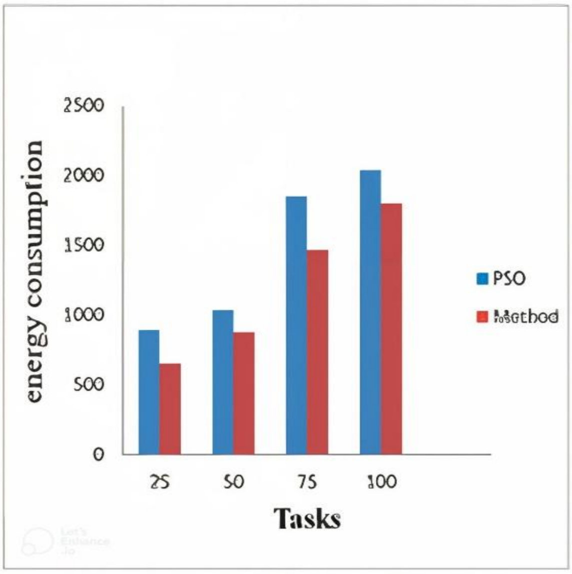

Figure 4: Comparison of energy consumption between PSO and proposed method

5. CONCLUSION

Cloud computing has become one of the leading technologies due to its unique ability to share computational resources and provide scalable services. However, the high energy consumption in data centers remains a significant challenge, leading to increased operational costs and environmental concerns. The aim of this study was to design and implement a new meta-heuristic algorithm that optimizes energy consumption while improving resource efficiency. The method of this research focused on simulating the proposed algorithm in real cloud environments and using reliable datasets such as Amazon EC2 and Planet Lab. In this process, the performance of the proposed algorithm was compared with traditional algorithms in terms of execution time, energy consumption, and resource efficiency. Advanced optimization techniques and precise mathematical modeling were also used to validate the results under real-world conditions to fully assess the generalizability of the proposed algorithm. The results showed that the proposed meta-heuristic algorithm significantly reduced energy consumption. Specifically, the average energy consumption of the proposed method was 45.7 kWh, which is 15.8% lower than the 54.3 kWh energy consumption of the PSO algorithm. Additionally, the proposed algorithm reduced task execution time by 14.6%, decreasing the average execution time from 960 seconds in the PSO algorithm to 820 seconds. This improvement was due to optimal resource allocation and fine-tuning of parameters. Moreover, resource efficiency increased by 10.8%, reaching 92.3%, compared to 81.5% in the PSO algorithm. These results confirm the effectiveness of the proposed algorithm in optimizing energy consumption, reducing task execution time, and increasing resource efficiency, making it a valuable tool for efficient resource management in cloud environments.

6. REFERENCES

[1] Al-Wesabi, F. N., Obayya, M., Hamza, M. A., Alzahrani, J. S., Gupta, D., & Kumar, S. (2022). Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment. Sustainable Computing: Informatics and Systems, 35, 100686.

[2] Singhal, S., Gupta, N., Berwal, P., Naveed, Q. N., Lasisi, A., & Wodajo, A. W. (2023). Energy efficient resource allocation in cloud environment using metaheuristic algorithm. IEEE Access, 11, 126135-126146.

[3] Abdel-Basset, M., El-Shahat, D., Elhoseny, M., & Song, H. (2020). Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications. IEEE Internet of Things Journal, 8(16), 12638-12649.

[4] Al-Mahruqi, A. A. H., Morison, G., Stewart, B. G., & Athinarayanan, V. (2021). Hybrid heuristic algorithm for better energy optimization and resource utilization in cloud computing. Wireless Personal Communications, 118, 43-73.

[5] Mirmohseni, S. M., Tang, C., & Javadpour, A. (2022). FPSO-GA: a fuzzy metaheuristic load balancing algorithm to reduce energy consumption in cloud networks. Wireless Personal Communications, 127(4), 2799-2821.

[6] Jomah, S. (2024, October). Meta-Heuristic Scheduling: A Review on Swarm Intelligence and Hybrid Meta-Heuristics Algorithms for Cloud Computing. In Operations Research Forum (Vol. 5, No. 4, p. 94). Cham: Springer International Publishing.

[7] R Zolfaghari, AM Rahmani, Virtual machine consolidation in cloud computing systems: Challenges and future trends, Wireless Personal Communications 115 (3), 2020, 2289-2326.

[8] R Zolfaghari, A Sahafi, AM Rahmani, R Rezaei, Application of virtual machine consolidation in cloud computing systems Sustainable Computing: Informatics and Systems 30, 2021, 100524.

[9] R Zolfaghari, A Sahafi, AM Rahmani, R Rezaei, An energy-aware virtual machines consolidation method for cloud computing: Simulation and verification, Software: Practice and Experience 52 (1), 2022, 194-235.

[10] Rahmat Zolfaghari, Energy-performance aware virtual machines migration in cloud network by using prediction and fuzzy approaches, Engineering Applications of

Artificial Intelligence 131 (2024) 107825.

[11] Prity, F. S., Uddin, K. A., & Nath, N. (2024). Exploring swarm intelligence optimization techniques for task scheduling in cloud computing: algorithms, performance analysis, and future prospects. Iran Journal of Computer Science, 7(2), 337-358.

[12] MA Khameneh, R Zolfagharim, Consensus Process in Large-Scale Group Decision-Making Based on Limited Trust and Social Networks Using Machine Learning, 2025 11th International Conference on Web Research (ICWR), 116-122.

[13] EA Abyaneh, R Zolfaghari, AA Abyaneh, User Intent Detection in Persian Text-Based Chatbots: A Comprehensive Review of Methods and Challenges, 2025 11th International Conference on Web Research (ICWR), 243-249.