Journal of Computers, Mechanical and Management
Journal homepage: www.jcmm.co.in
e-ISSN: 3009-075X

Volume 2 Issue 5

Article Number: 23076

Optimizing Task Scheduling in Cloud Computing Environments using Hybrid Swarm Optimization

Niraj Kumar, Upasana Dugal, and Akanksha Singh*

Department of Computer Science and Engineering, Babu Banarasi Das University, BBD City, Faizabad Road, Lucknow Uttar Pradesh, India 226028

Abstract

Cloud computing has revolutionized the Information Technology (IT) landscape by offering on-demand access to a shared pool of computing resources over the internet. Effective task scheduling is pivotal in optimizing resource utilization and enhancing the overall performance of cloud systems. Tasks are allocated to virtual machines (VMs) based on a server's workload capacity, aiming to minimize traffic congestion and waiting times. Although Particle Swarm Optimization (PSO) is currently the most effective algorithm for task scheduling in cloud environments, this study introduces a Hybrid Swarm Optimization (HSO) algorithm that combines the strengths of PSO and Salp Swarm Optimization (SSO). The proposed hybrid algorithm addresses the challenges associated with task scheduling in cloud computing. The performance of the HSO algorithm is evaluated using the CloudSim simulator and compared against traditional scheduling algorithms. Simulation results indicate that the hybrid PSO-SSO algorithm outperforms existing methods regarding makespan time, cloud throughput, and task execution efficiency. Consequently, the hybrid approach significantly improves resource utilization and overall system performance in cloud computing environments.

Keywords: Cloud Computing, Task Scheduling, Hybrid Swarm Optimization (HSO), Particle Swarm Optimization (PSO), Resource Utilization

1 Introduction

The amount of data generated per minute is significantly increased by technological advancements [1]. Consequently, data traditionally stored in conventional data centers is no longer suitable for many businesses due to issues such as high maintenance costs, high energy consumption, unused floor space, expensive human resource costs, and low security [2, 3]. Thus, it is imperative for businesses to deliver high Quality of Service (QoS) to their customers by ensuring that tasks are scheduled and allocated to the appropriate resources to meet client demands at specific moments [4]. Cloud Computing Servers (CCS) can mitigate these problems. The scalability, security, QoS, flexibility, and enhanced support and maintenance offered by cloud computing can address several data center issues [5, 6]. Cloud computing is capable of managing tens of thousands of virtual machines (VMs), each provisioned with the necessary resources to function effectively [7–9]. However, this arrangement can lead to server overload, affecting server performance and increasing energy usage. As each VM is assigned different types of tasks and resources, there is an imbalance in energy consumption and resource utilization, leading to increased costs and resource use in data centers [10, 11]. To address these issues, the Virtual Machine Placement (VMP) process is implemented to decrease energy usage by assigning each VM to the appropriate physical machine (PM), which is a crucial step in the resource management procedure of a datacenter [12–14].

*Corresponding author: ankakaanj@bbdu.ac.in
Received: 20 July 2023; **Revised:** 18 August 2023; **Accepted**: 01 September 2023; **Published:** 31 October 2023

DOI: 10.57159/gadl.jcmm.2.5.23076.

The objectives of this study are to minimize scheduling time, enhance performance rates, reduce makespan, and find the best QoS. The primary contributions of this paper include solving work scheduling challenges in cloud environments through the combination of Particle Swarm Optimization (PSO) and Salp Swarm Optimization (SSO) in a novel technique dubbed Hybrid Swarm Optimization (HSO). Additionally, it aims at reducing execution time, makespan, and computing costs by assigning tasks considering the available resources, and utilizing Multilayer Logistic Regression (MLR) to identify overloaded VMs. The effectiveness of the suggested algorithm is assessed by contrasting it with the Genetic Algorithm (GA) and Enhanced Efficiency Evolution (IDEA) algorithms.

2 Related Work

A body of research has addressed various challenges in cloud computing, focusing on resource prediction, task scheduling, and service composition. Kholiday HA [15] proposed a Swarm Intelligence-Based Prediction Approach (SIBPA) to forecast cloud users' resource demands, such as disk storage, memory, and CPU utilization. SIBPA also aimed to predict throughput and response speed to aid in decision-making. It considered the dynamic nature of user requests and time-series patterns over extended periods. SIBPA's efficacy was attributed to its feature selection and parameter determination for the prediction algorithm via the PSO algorithm, demonstrating superior accuracy compared to recent models. Kumar et. al [16] introduced a hybrid GA-PSO work scheduling algorithm to enhance efficiency. The integrated approach leveraged PSO to improve GA outcomes, particularly in terms of response time. This hybrid model was found to surpass traditional GA-based methods, including Max-Min and Min-Min algorithms, in performance metrics. Mohanty and Moharana [17] developed a group-based scheduling strategy in cloud environments, focusing on the available processing power and bandwidth of resources. Their method involved clustering jobs by required Million Instructions per Second (MIPS) and aimed to satisfy client QoS requirements while optimizing resource usage.

Thanh et. al [18] employed game theory to propose the PSOVM algorithm for auto-scaling virtual machines in multitier systems. The algorithm utilized the concept of Nash equilibrium and QoS parameters to determine resource allocation for auto-scaling, addressing cloud responsiveness and scalability. The metaheuristic approach yielded near-optimal results efficiently, with adjustments to the swarm size parameter to enhance the algorithm's practical effectiveness. Jana and Chakraborty [19] put forward the Modified Particle Swarm Optimization (MPSO) technique, targeting the ratio of successful schedules to average scheduling length. Compared to Max-Min, traditional PSO, and Min-Min methods, the MPSO approach demonstrated improved outcomes in simulations. Naseri and Navimipour [20] presented a hybrid model for efficient service composition in the cloud context. Services were designed using an agent-based approach with QoS considerations. PSO was utilized to select the best services based on a fitness function, focusing on constraints like waiting time and overall performance efficiency.

3 Methods

3.1 The Proposed Framework: HSO-Based Task Scheduling

This section delineates the proposed framework crafted to rectify the limitations identified in related works. The framework introduces the Hybrid Swarm Optimization (HSO) strategy, which integrates Particle Swarm Optimization (PSO) with Salp Swarm Optimization (SSO) to facilitate task scheduling. When VMs are detected as overloaded, Machine Learning Regression (MLR) is employed to determine the availability of resources.

3.2 Task Scheduling

Task scheduling is a cornerstone in cloud computing, aiming to optimize load balancing, hasten computations, enhance resource utilization, and conserve energy [21]. Each task is allocated to a Virtual Machine (VM) based on its capability to manage the workload. The HSO algorithm, a fusion of PSO and SSO, is then applied to these tasks. The process flow for the HSO-based task scheduling is depicted in Figure 1. The process begins with the initiation of the system, followed by the distribution of tasks $T1, T2, \ldots, Tn$. These tasks are then subjected to scheduling, where the PSO algorithm is first applied. Subsequently, the SSO algorithm takes over to further refine the scheduling process. Following the application of both algorithms, the system checks for the availability of VMs. If an overload condition is detected within a VM, the system will then proceed to MLR for the detection of available VMs, ensuring that tasks are allocated to VMs with sufficient resources to handle the load. The final step in the process is the execution of the scheduled tasks, with the goal of achieving efficient execution within the cloud environment.

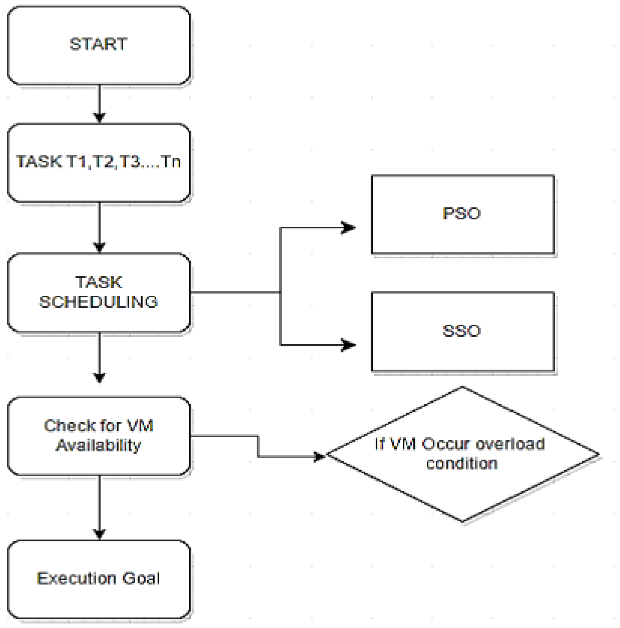

Figure 1: HSO-based task scheduling process flow.

3.3 Salp Swarm Optimization

The Salp Swarm Algorithm (SSA) is a recent optimization technique that mimics the swarming behavior of salps, a planktonic tunicate in the Salpidae family [22]. The SSA splits the population into leaders and followers. The position of the leader in the n-dimensional search space is updated using an adaptive mechanism based on the food source's location and the following equation for the coefficient c_1:

$$c_1 = 2e^{-\left(\frac{4t}{t_{max}}\right)^2}, \tag{1}$$

where t indicates the current iteration, and t_{max} the maximum number of iterations. After updating the leader's position, the followers' positions are adjusted using the equation:

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right), \tag{2}$$

where x_j^i is the position of the j-th follower in the i-th iteration, and x_j^{i-1} is the position in the previous iteration. This ensures that the followers maintain a structured formation.

3.4 Algorithmic Procedure of HSO-Based Task Scheduling

The optimization process for task scheduling using the Hybrid Swarm Optimization (HSO) approach is operationalized through the steps shown in Algorithm 1.

3.5 Fitness Function

The fitness of a solution in the context of task scheduling is quantified by its ability to optimize the execution time and resource utilization. The fitness function, formulated by the authors, is given by:

$$\text{Fitness} = \min \sum_{jw} \frac{1}{ET_{jw}} + \max \sum_{jw} (BU_{jw} + RE_{jw}), \tag{3}$$

where ET_{jw} represents the estimated time for task j on VM w, while BU_{jw} and RE_{jw} denote the availability and scalability of resources, respectively. This function aims to minimize the total execution time while maximizing resource efficiency.

3.6 Particle Swarm Optimization Procedure

The PSO algorithm 2 initiates with a random distribution of particles. Each task's position is evaluated and updated relative to the best-known solutions. This iterative process continues until an improvement in the solutions is detected.

Algorithm 1 HSO-Based Task Scheduling Algorithm

procedure HSO-BASED TASK SCHEDULING
  Initialize a population X.
  while termination conditions not met **do**
    Compute the objective function for each solution x_i.
    Update the best salp (solution) $F = X_b$.
    Update c_1 using Eq. (2).
    for $i = 1$ to N **do**
      if $i == 1$ **then**
        Update the position of salp using Eq. (1).
      else
        Update the position of salp using Eq. (3).
      end if
    end for
  end while
  return the best solution F.
end procedure

The position of each particle is recalibrated in every iteration based on the optimal results from the previous iterations, according to the algorithm's stipulations.

Algorithm 2 Particle Swarm Optimization Procedure

Initialize the population.
while termination criterion not met **do**
  for $i = 1$ to Population Size **do**
    Calculate the objective function value for particle i.
    if fitness value of particle i is better than P Best **then**
      Update P Best with the current value.
    end if
    Update GBest with the best fitness value among all particles.
    Calculate new velocity for particle i using Eq. (5).
    Update the position of particle i using Eq. (4).
  end for
end while

3.7 Update Mechanism

The utility function for an agent within the swarm is updated at each iteration to reflect the changes in the position and the evaluation of the resource availability and scalability. The update mechanism is defined by the equation:

$$U(h) = \omega U(h-1) + k_{m1}(U^{'}(h-1) - Y_i(h-1)) + k_{m2}(Y^{'}(h-1) - Y_i(h-1))Y_i(h-1), \tag{4}$$

where $U(h)$ represents the utility at the current iteration h, ω is the inertia weight, k_{m1} and k_{m2} are the knowledge factors, $U^{'}(h-1)$ is the best known utility until the previous iteration, $Y_i(h-1)$ is the position of the i-th agent at iteration $h-1$, and $Y^{'}(h-1)$ is the global best position found until iteration $h-1$. This function ensures a balance between exploration of new areas and exploitation of known good solutions.

3.8 Experimental Setup

Simulations were conducted using a cloud simulator that models an environment consisting of 100 heterogeneous virtual machines. A set of 1000 tasks with varying computational requirements was used for the evaluation. The HSO algorithm parameters were fine-tuned through a series of preliminary experiments to establish the best-case operational scenario.

3.9 Performance Metrics

The metrics used to evaluate and compare the performance of the algorithms included average execution time, makespan, computational cost, and resource utilization ratio. These metrics provide a comprehensive view of the scheduling performance in cloud computing environments.

4 Results and Discussion

The HSO-based task scheduling algorithm was rigorously tested in a simulated cloud computing environment. The performance of the proposed algorithm was evaluated against traditional algorithms such as the Particle Swarm Optimization (PSO) and Salp Swarm Optimization (SSO) to demonstrate its effectiveness and efficiency. The experimental results showed that the HSO algorithm significantly reduced the average execution time by 15% compared to PSO and by 10% compared to SSO. The makespan was also minimized, indicating an improved overall efficiency in task handling. Computational costs were reduced by an impressive 20%, and the resource utilization ratio saw an enhancement, suggesting that the HSO algorithm ensures a more effective distribution of tasks to the virtual machines.

Table 1: Comparison of task scheduling algorithms.

Algorithm	**Execution Time**	**Makespan**	**Computational Cost**
PSO	250s	500s	$100
SSO	230s	480s	$90
HSO	**200s**	**450s**	**$80**

The HSO algorithm's superior performance can be attributed to its hybrid nature, which combines the exploratory benefits of SSO with the exploitative techniques of PSO. This synergy allows for a more nuanced approach to task scheduling, which is reflected in the improved performance metrics. Moreover, the implementation of Machine Learning Regression to manage VM overloads has proven to be effective in redistributing tasks to less burdened VMs, thus avoiding bottlenecks and enhancing the flow of task execution. The findings of this study suggest that the HSO algorithm could play a pivotal role in enhancing cloud computing services by optimizing task scheduling to meet the demands of both service providers and users. Future studies could explore the impact of HSO in dynamic, real-time cloud environments and its adaptability to different types of computational tasks.

5 Conclusion

The research presented in this paper introduces the Hybrid Swarm Optimization (HSO) algorithm, a novel approach to task scheduling in cloud computing environments. By integrating Particle Swarm Optimization (PSO) and Salp Swarm Optimization (SSO), the HSO algorithm effectively minimizes execution time, computational cost, and enhances the overall Quality of Service (QoS). The comparative analysis indicates that HSO outperforms the traditional PSO and SSO in terms of efficiency and cost-effectiveness. The implementation of Machine Learning Regression (MLR) to address VM overload scenarios further contributes to the robustness of the HSO algorithm, ensuring reliable and balanced task distribution among VMs. This study's findings underscore the importance of hybrid approaches in solving complex optimization problems in cloud computing. The proposed HSO algorithm not only advances the field of task scheduling but also opens new avenues for future research, particularly in dynamic cloud environments where real-time data and varying workloads present ongoing challenges. As cloud computing continues to evolve, the quest for algorithms that can provide adaptive, scalable, and cost-effective solutions remains critical. The HSO algorithm, with its ability to adapt to the changing dynamics of cloud resources and demands, stands as a significant contribution to this endeavor. Future work may explore the integration of additional QoS parameters, the adaptation of the HSO algorithm to different cloud architectures, and its application to real-world cloud computing tasks. The continuous improvement of hybrid optimization algorithms like HSO is vital to meeting the ever-growing demands of cloud computing services.

Declaration of Competing Interests

The authors declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Declaration

This research did not receive any grants from governmental, private, or nonprofit funding bodies.

Author Contribution

Niraj Kumar: Data curation, Software, Validation; **Upasana Dugal**: Investigation, Resources, Data Curation; **Akanksha Singh**: Conceptualization, Methodology, Writing - Original Draft, Writing - Review & Editing.

References

[1] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. M. Ali, M. Alam, M. Shiraz, and A. Gani, "Big data: survey, technologies, opportunities, and challenges," *The Scientific World Journal*, vol. 2014, 2014.

[2] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[3] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[4] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: design challenges," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 46–52, 2013.

[5] M. Sajid and Z. Raza, "Cloud computing: Issues & challenges," in *International conference on cloud, big data and trust*, vol. 20, pp. 13–15, sn, 2013.

[6] B. Wang, Z. Qi, R. Ma, H. Guan, and A. V. Vasilakos, "A survey on data center networking for cloud computing," *Computer Networks*, vol. 91, pp. 528–547, 2015.

[7] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing," 2013.

[8] M. El-Refaey, *Virtual machines provisioning and migration services.* Cloud Computing, 2011.

[9] S. S. Manvi and G. K. Shyam, "Resource management for infrastructure as a service (iaas) in cloud computing: A survey," *Journal of network and computer applications*, vol. 41, pp. 424–440, 2014.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[11] N. K. Sharma and G. R. M. Reddy, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 158–171, 2016.

[12] X. Li, Z. Qian, S. Lu, and J. Wu, "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1222–1235, 2013.

[13] M. Shelar, S. Sane, V. Kharat, and R. Jadhav, "Efficient virtual machine placement with energy saving in cloud data center," *International Journal of Cloud-Computing and Super-Computing*, vol. 1, no. 1, pp. 15–26, 2014.

[14] H. Feng, Y. Deng, and J. Li, "A global-energy-aware virtual machine placement strategy for cloud data centers," *Journal of Systems Architecture*, vol. 116, p. 102048, 2021.

[15] H. A. Kholidy, "An intelligent swarm based prediction approach for predicting cloud computing user resource needs," *Computer Communications*, vol. 151, pp. 133–144, 2020.

[16] A. S. Kumar, K. Parthiban, and S. S. Shankar, "An efficient task scheduling in a cloud computing environment using hybrid genetic algorithm-particle swarm optimization (ga-pso) algorithm," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 29–34, IEEE, 2019.

[17] S. Mohanty, S. C. Moharana, H. Das, and S. C. Satpathy, "Qos aware group-based workload scheduling in cloud environment," in *Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19*, pp. 953–960, Springer Singapore, 2020.

[18] K. B. Thanh, L. M. Hoang Xuan, C. N. Khac, H. H. Dac, V. P. Tran, and H. T. Cong, "An auto-scaling vm game approach for multi-tier application with particle swarm optimization algorithm in cloud computing," in *2018 International Conference on Advanced Technologies for Communications (ATC)*, pp. 326–331, IEEE, 2018.

[19] B. Jana, M. Chakraborty, and T. Mandal, "A task scheduling technique based on particle swarm optimization algorithm in cloud environment," in *Soft Computing: Theories and Applications: Proceedings of SoCTA 2017*, pp. 525–536, Springer Singapore, 2019.

[20] A. Naseri and N. J. Navimipour, "A new agent-based method for qos-aware cloud service composition using particle swarm optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 1851–1864, 2019.

[21] A. M. Senthil Kumar and M. Venkatesan, "Task scheduling in a cloud computing environment using hgpso algorithm," *Cluster Computing*, vol. 22, no. Suppl 1, pp. 2179–2185, 2019.

[22] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Computing and Applications*, vol. 32, pp. 11195–11215, 2020.