Response Time Enhancement for Task Scheduling in Cloud Computing

Nora Omran Alkaam*

Iraqi Ministry of Higher Education and Scientific Research, Iraq

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.101000236

Received: 19 October 2026; Accepted: 24 October 2026; Published: 01 November 2026

ABSTRACT

Cloud computing has become the primary method for providing on-demand computing services, and provides both flexibility and scalability for many types of applications. The efficiency of cloud environments, however, depends heavily upon the effectiveness of the scheduling of tasks, which is a known NP-complete problem. A good scheduler should find ways to achieve a number of different goals that conflict with each other (such as; reducing makespan, increasing resource usage, and decreasing response times) in order to obtain the best possible performance. Recent research has turned to hybrid meta-heuristic algorithms as potential solutions to this very difficult problem. The Henry Gas-Harris Hawks Modified Opposition (HGHHM) algorithm was first developed to address cloud task scheduling. Since then it has proven to be an effective tool for improving makespan, resource utilization, and degree of imbalance. This paper takes that work further, and evaluates HGHHM in terms of its ability to reduce response times for users. Experiments were run using the CloudSim toolkit, with the HGHHM compared against the eDSOS benchmark algorithm in different workload sizes.

Results indicate that HGHHM outperforms eDSOS in minimizing response time, and also has been shown to be superior to eDSOS in terms of resource utilization and makespan. Statistical testing verified the significance of this difference. The experimental findings show that for the normal, left-half, right-half, and uniform task-size distributions, HGHHM algorithm achieves PIR values of 98.39%, 97.96%, 97.60%, and 97.63%, respectively. The algorithm's efficacy under various task-size characteristics is demonstrated by these findings, which show consistently high PIR values across all four workload distributions. Thus, these results evaluate HGHHM more thoroughly and illustrate the versatility of HGHHM as an effective hybrid meta-heuristic for scheduling tasks on clouds.

Keyword: cloud computing, response time, Harris hawks, Henry gas, optimization.

INTRODUCTION

Cloud computing is widely recognized as a transformative paradigm that integrates diverse resources including storage, processors, networks, memory, and applications—delivered as on-demand services, alongside dynamically evolving IT infrastructure[1]. By adopting this model, organizations significantly reduce the costs associated with software and hardware acquisition, application deployment, and system maintenance. Moreover, the inherent scalability of cloud environments mitigates the risks of inaccurate service-scale estimation, which can otherwise lead to resource wastage in cases of over-provisioning or revenue loss in cases of under-provisioning[2,3]. Cloud computing enables resource sharing among users through virtualization, a technique that allows multiple remote environments to be safely consolidated onto physical machines. This process ensures optimal utilization of physical resources and energy[2]. Cloud data are distributed across interconnected servers, which are accessed through virtual machines (VMs). As a result, VMs have become a critical component of cloud data centers, serving as the foundation for software stacks and resource management.

Within this context, efficient task scheduling strategies are essential for optimizing cloud performance. These

strategies aim to minimize computational metrics such as makespan, response time, system throughput, and system utilization, as well as network-oriented metrics including traffic volume, communication cost, data transfer cost, and round-trip time[4]. Performance metrics such as these are critical to evaluating how well a cloud is operating and what needs to be improved in areas like Quality of Service (QoS), Service Level Agreement (SLA) Compliance, Load Balancing, Energy Efficiency, and Fault Tolerance[4,5]. In recent years, although there has been a lot of work done on task scheduling in the cloud computing area, the task scheduling issue is still an NP-complete problem that cannot be solved exactly. For the most part, traditional task scheduling algorithms used in the cloud are rule-based. This is primarily because of how easy it is to implement rules. Unfortunately, however, rule-based algorithms have been found to perform very poorly when they are used with complex scheduling situations[6-11].

To address these limitations, recent studies have increasingly employed metaheuristic optimization techniques, which have demonstrated superior performance compared to conventional heuristics or local search methods. Widely used metaheuristics in cloud task scheduling include Rock Hyrax Algorithm[12], Elite learning Harris hawks optimizer[13], Ant Colony Optimization[14,15]. Genetic Algorithms[16]. To evaluate the proposed strategy, four types of task size distributions: normal, left-skewed, right-skewed, and uniform, were employed following the experimental setup in[17]. This design enabled a comprehensive assessment of performance under varied workload conditions. The rationale for using diverse distributions lies in their ability to capture realistic application scenarios. For instance, many long-running applications such as web services exhibit fluctuating CPU requirements, while high-performance computing (HPC) applications, though short-lived, exert substantial computational pressure on CPUs. The adopted distributions therefore represent distinct scheduling contexts: the normal distribution models a balanced mix of web and HPC tasks; the left-skewed distribution reflects environments where HPC tasks dominate; the right-skewed distribution represents scenarios with a higher likelihood of web tasks; and the uniform distribution depicts cases in which a single type of application is scheduled. Collectively, these experimental settings provide valuable insights into the adaptability and robustness of the proposed eDSOS approach[17].

Building upon this foundation, the present study extends the previously introduced Henry Gas–Harris Hawks Modified-Opposition (HGHHM) algorithm, which had been applied to minimize makespan, maximize resource utilization, and reduce the degree of imbalance in cloud scheduling[5]. In this work, the focus shifts toward response time as a primary optimization objective, thereby providing a more comprehensive evaluation of HGHHM's effectiveness in enhancing overall scheduling performance in cloud computing environments. The remainder of this paper is structured as follows. Next Section reviews related work. Section III presents the Mathematical Model of the Objective Function. Section IV outlines the experimental environment. Section V show the results and their analysis. Finally, Section VI concludes the paper and highlights potential directions for future research.

Related Works

Cloud computing is now a major model of providing virtualized computing resources, and the performance of these systems depends heavily upon efficient scheduling of tasks and distribution of workloads. There are many ways that people have tried to improve how well the cloud performs by reducing makespan, improving response times and increasing resource utilization. Sa et al. developed an enhanced version of a discrete symbiotic organism search algorithm using meta-heuristics for scheduling jobs in cloud computing environments. Their method outperformed benchmark techniques in large search spaces, achieving notable improvements in makespan and response time[17]. However, the approach suffered from frequent convergence to local optima, limiting scalability and robustness[17]. Similarly, Singhal et al.proposed a nature-inspired job scheduling approach based on the Rock Hyrax Optimization Algorithm to improve resource allocation in cloud computing environments. Their method integrates the K-means clustering algorithm to organize cloud resources into clusters according to their computational, storage, and network capabilities. The experimental results indicated improvements in scheduling performance, including reductions in makespan and energy consumption with shorten response time compared with the evaluated approaches[18].

Ledmi proposed a Discrete Tuna Swarm Optimization for Task Scheduling (DTSO-TS) algorithm, adapted from the original Tuna Swarm Optimization (TSO) technique. The algorithm aims to optimize task allocation

across virtual machines by balancing workloads and improving resource utilization, with the objective of reducing makespan and response time while increasing throughput[19]. Devaraj, proposed a hybrid load-balancing algorithm that combines the Firefly algorithm with Improved Multi-Objective Particle Swarm Optimization to optimize the search process and improve task scheduling. Simulation results demonstrated improvements in response time, CPU and memory utilization, reliability, throughput, and makespan compared with other evaluated methods. The algorithm achieved an average response time of 13.58 ms, CPU utilization of 98%, and memory utilization of 93% [20]. Ebadifard et al. proposed a dynamic task-scheduling method for cloud computing that incorporates load balancing to improve response time and resource utilization. The approach distributes tasks fairly among virtual machines and considers their previous performance to enhance scheduling reliability. Simulation results demonstrated improvements in load balancing and reliability compared with existing methods[21]In addition to this, Zhou et al. also improved the effectiveness of the Genetic Algorithm (GA) as a means of improving scheduling efficiency through the use of a greedy approach. This allowed their approach to rapidly reach the desired solution in just a few iterations; resulting in lower makespans and response times for users, all the while providing acceptable quality of service (QoS) from the users point of view[22.]

Even though there have been some significant developments in the area of task scheduling and load balancing algorithms; however, further improvements in response time are required, especially when dealing with varying workloads and dynamically changing cloud environments. While many current methods for task scheduling/load balancing primarily focus on reducing makespan, maximizing resource usage, or achieving load balancing; response time has yet to be thoroughly examined. The limitations of these existing methods demonstrate the need for improved adaptive/efficient optimization techniques which will allow us to achieve shorter response times, while still allowing us to effectively utilize resources and maintain a balanced workload.

Mathematical Model of the Objective Function

The cloud task-scheduling problem is formulated using objective functions that evaluate the quality of candidate scheduling solutions. These functions guide the optimization algorithm toward finding near-optimal solutions. Although the HGHHM algorithm was previously developed to address multiple scheduling objectives, the present study focuses specifically on minimizing response time. When a cloud broker (CB) receives a set of tasks $\{T_1, T_2, T_3, T_4 \dots \dots, T_n\}$ the cloud information service (CIS) identifies the required services and maps them to a set of heterogeneous virtual machines (VMs) $\{V_1, V_2, V_3, V_4 \dots \dots, V_m\}$, Since VMs differ in processing speed and capacity, executing the same task on different VMs can lead to varying completion times.

The Expected Time to Compute (ETC) for task (i) on VM (j) is defined as:

$$ETC_{i,j} = \frac{\text{Task Length}_i}{\text{VM Power}_j}$$

where (VMPower_j) represents the processing speed of VM (j) in MIPS.

The Completion Time (CT) of a VM is the total execution time of all assigned tasks:

$$CT_i = \sum_{j=1}^{n} ETC_{i,j}$$

while its submission time is denoted by (ST). Accordingly, its response time is defined as:

$$RT_i = CT_i - ST_i$$

Finally, the fitness function evaluates the quality of each scheduling solution:

[Fitness = min { RT∀ tasksj mapped to VM i } i ∈ 1, 2, 3, . .. , m

This formulation ensures that task assignments minimize response time while balancing workloads across available VMs[16].

Experimental Environment

In this section, I am going to provide details about my experimental setup; specifically, I will describe the dataset used in the experiments and the parameters that were set in the experiment. I am going to use CloudSim, which is a popular open source simulation toolkit designed for modeling and simulating cloud computing environments, to assess how well the HGHHM algorithm performs when it comes to response time. CloudSim supports many of the most important elements of cloud computing infrastructure including data centers, hosts, VMs, cloud service brokers, and resource allocation policies.23

The experiments took place within the context of a typical desktop computer system running CloudSim toolkit v3.0.3. There was one data center modeled, which contained two host computers that supported a total of 20 virtual machines. The computational ability of each virtual machine is expressed in terms of MIPS (Million Instructions Per Second). In addition, a synthesized dataset was employed to evaluate the response time of the HGHHM algorithm. The workloads were modeled as non-preemptive and independent tasks, ensuring fair evaluation of scheduling efficiency. To capture diverse workload behaviors, task traces were generated using four distribution types: normal, uniform, left-skewed, and right-skewed. To ensure reliability and accuracy of the results, each experiment was independently repeated for approximately 1000 cycles per technique, and the outcomes were averaged over multiple iterations[17,23]. This iterative evaluation strategy helped mitigate random fluctuations and ensured statistically meaningful results.

RESULTS AND DISCUSSION

The response time metric measures the efficiency of the proposed HGHHM task scheduling strategy in processing tasks across heterogeneous virtual machines. A lower response time value indicates faster execution and, therefore, superior scheduling performance. Figures 1- 4 illustrate the response time behavior of the HGHHM algorithm, highlighting the minimum response times achieved for tasks under different workload distributions.

The results clearly show that the proposed HGHHM consistently achieves the lowest response time for all tested datasets. This improvement can be attributed to the hybridization of the Harris Hawks Optimization (HHO) exploitation phase and the Henry Gas Solubility Optimization (HGSO) exploration phase. By effectively balancing local and global search processes, the HGHHM enhances decision-making for task-to-VM allocation, thereby avoiding resource underutilization and minimizing execution delays.

Additionally, the convergence behavior of HGHHM is shown in Figure 5, which shows how its best-so-far fitness values change over the course of subsequent cycles. The convergence curve suggests that HGHHM can preserve the best solutions discovered while still exploring the search space since it shows that the algorithm gradually improves its solutions before arriving at a stable optimization stage. This convergence pattern adds to the response-time statistics and sheds further light on how the suggested approach is optimized.

Another key observation is that the performance of the HGHHM scheduling strategy is influenced by the number of virtual machines. While variation exists across datasets, the algorithm maintains robust improvements in minimizing response times. Importantly, even for average response time values, HGHHM consistently outperforms the benchmarked enhanced discrete symbiotic organism search (eDSOS) algorithm.

To validate these results, a paired t-test was conducted to statistically compare the performance of HGHHM and eDSOS. The results, presented in Tables 1- 4, confirm that all p-values are below the 0.05 significance level, thereby rejecting the null hypothesis. This provides strong evidence that the improvements achieved by HGHHM are statistically significant across all dataset distributions.

Furthermore, the Performance Improvement Rate (PIR%)**,** summarized in Table 5, demonstrates the magnitude of enhancement achieved by the proposed strategy. Specifically, HGHHM reduces response time by 98.39%, 97.96%, 97.60%, and 97.63% across the four dataset types compared to eDSOS. These results confirm the effectiveness of HGHHM in improving both resource utilization and execution efficiency in cloud environments.

In summary, the findings highlight HGHHM's scalability and robustness in handling diverse workload distributions. By minimizing response time, the proposed hybrid approach demonstrates its potential as a versatile and highly efficient task scheduling solution for cloud computing. Most importantly, the significant reductions in response time directly contribute to improving Quality of Service (QoS) and meeting Service Level Agreement (SLA) requirements, reinforcing the practical relevance of the proposed approach in real-world cloud applications.

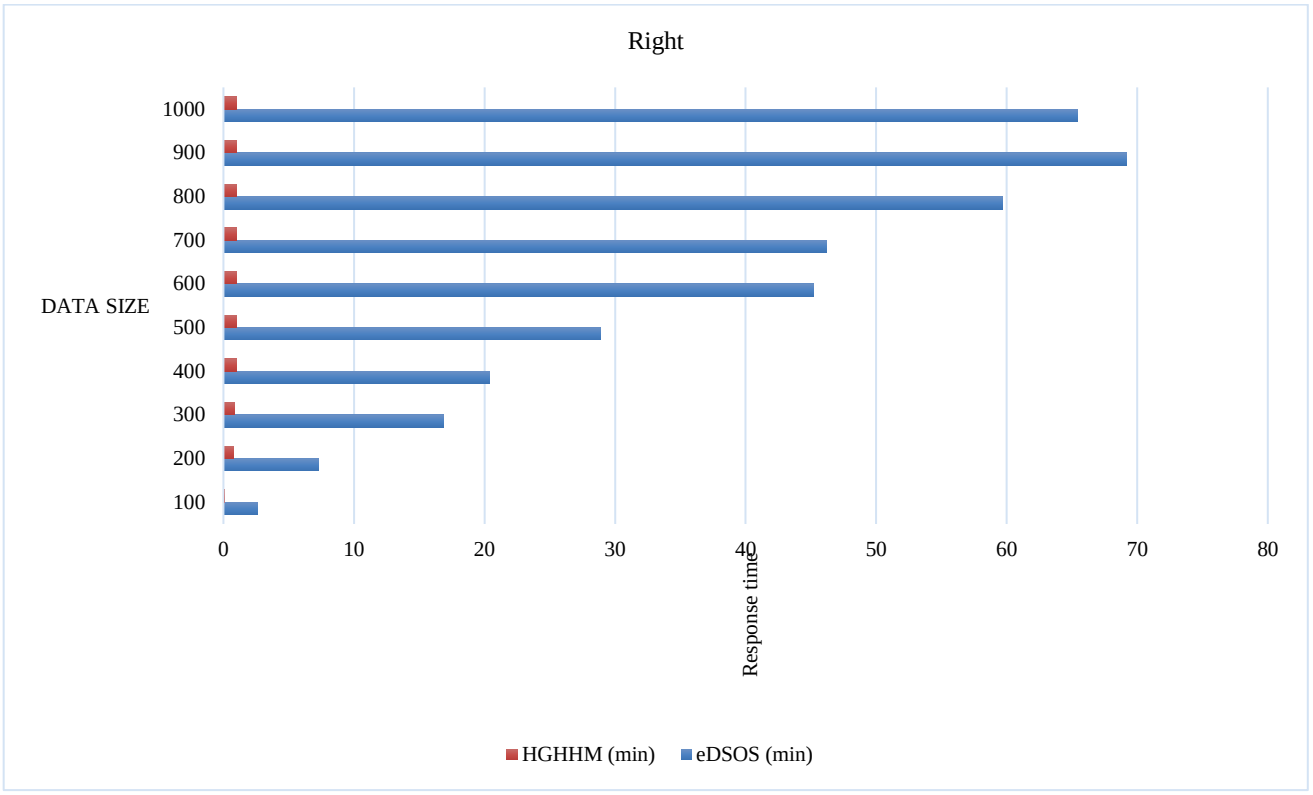


Figure 1: Response Time for Right Distribution Dataset

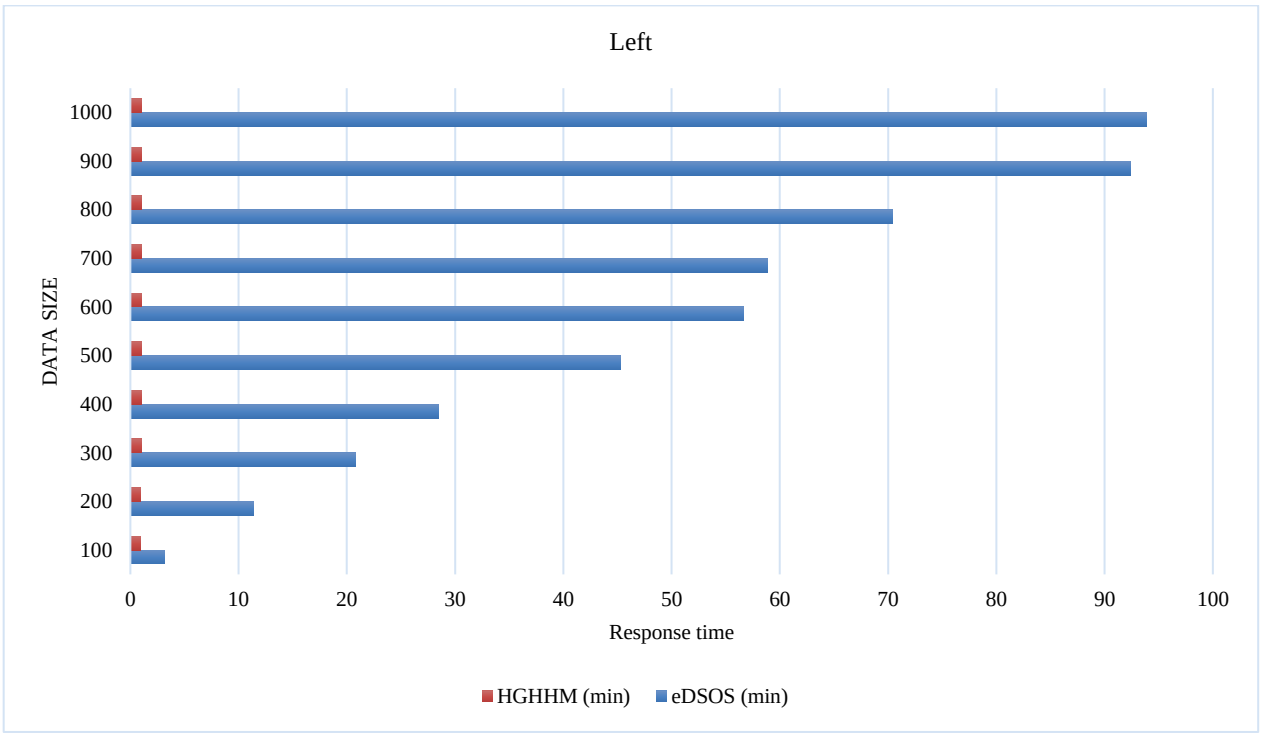


Figure 2: Response Time for Left Distribution Dataset

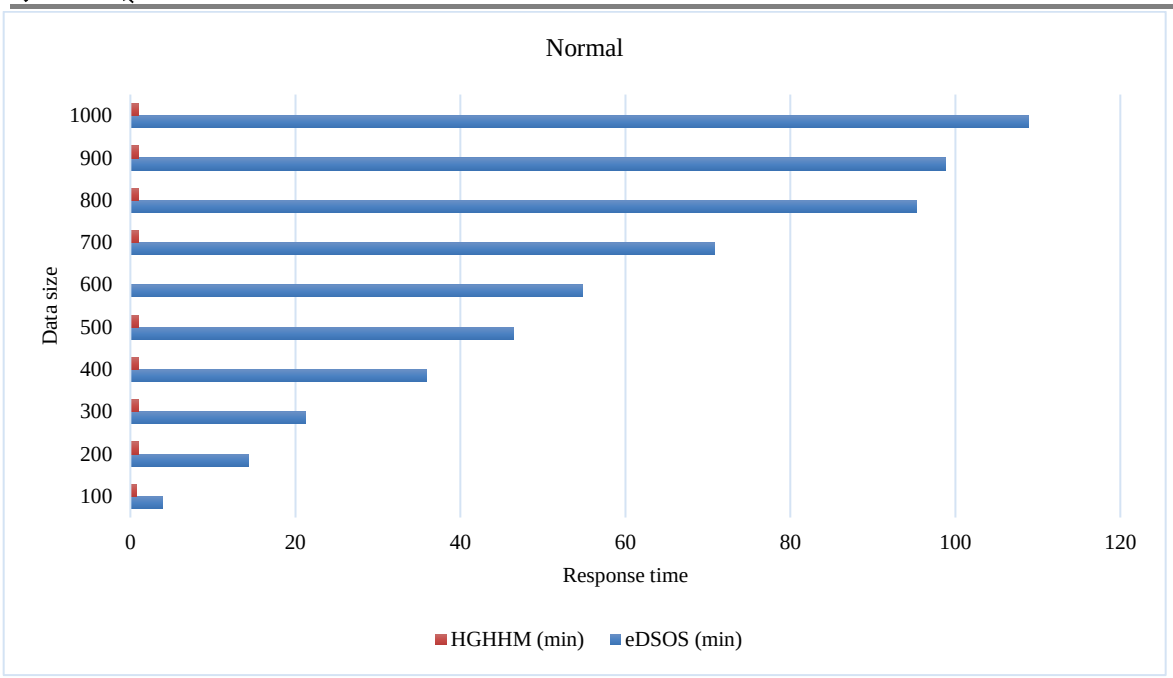


Figure 3: Response Time for Normal Distribution Dataset

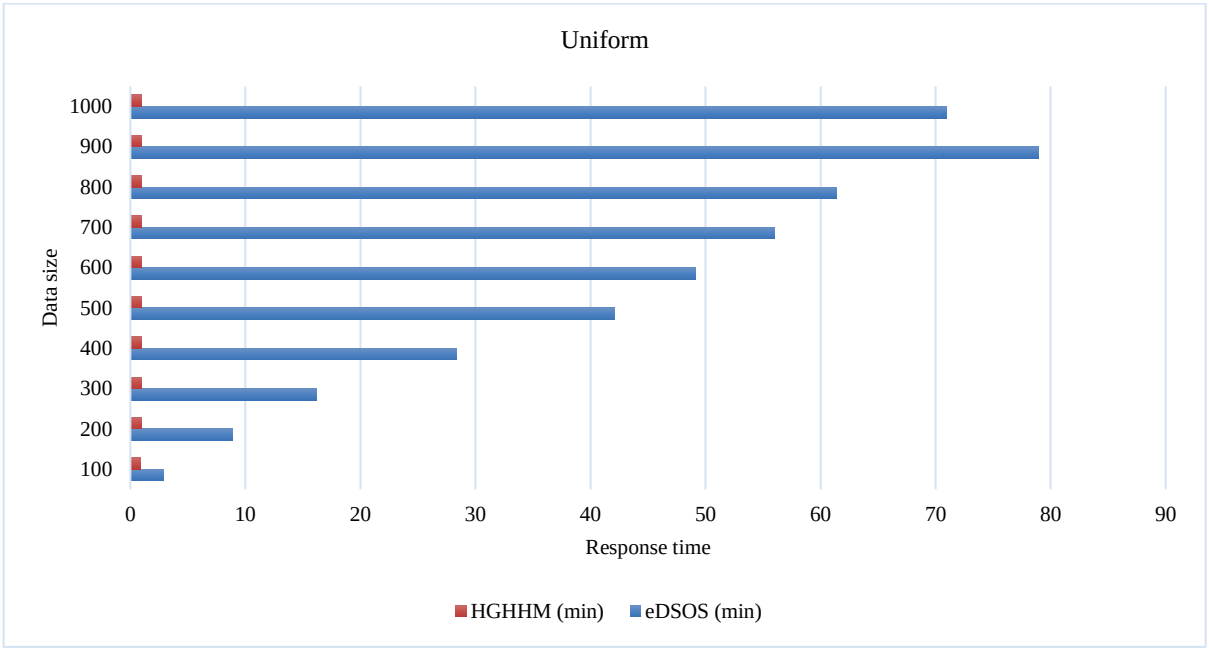


Figure 4: Response Time for Uniform Distribution Dataset

Table 1: Comparison of RT obtained by eDSOS and HGHHM for data instances generated using Normal distribution dataset

Task size	eDSOS			HGHHM			T-test
	Max	Min	Avg.	Max	Min	Avg.	
100	18.64	3.92	11.23	0.95	0.80	0.92	t-value=5.64
200	35.9	14.37	24.29	1.00	1.00	1.00	p-value≈0.0003
300	51.67	21.25	38.52	1.00	1.00	1.00	
400	69.61	35.91	54.13	1.00	1.00	1.00	
500	97.62	46.42	73.71	1.00	1.00	1.00	
600	97.73	54.86	79.75	1.00	0.05	0.91	

700	128.19	70.83	96.23	1.00	1.00	1.00	
800	136.78	95.24	113.61	1.00	1.00	1.00	
900	152.63	98.8	126.87	1.00	1.00	1.00	
1000	183.95	108.93	141.32	1.00	1.00	1.00	

Table 2: Comparison of RT obtained by eDSOS and HGHHM for data instances generated using Right distribution dataset

Task size	eDSOS			HGHHM			T-test
	Max	Min	Avg.	Max	Min	Avg.	
100	10.61	2.62	6.47	1.00	0.05	0.83	t-value=5.32
200	20.73	7.28	14.68	1.00	0.80	0.93	p-value≈ 0.0005
300	34.1	16.86	25.27	1.00	0.85	0.97	
400	49.66	20.39	36.87	1.00	1.00	1.00	
500	55.33	28.91	44.14	1.00	1.00	1.00	
600	63.74	45.18	56.93	1.00	1.00	1.00	
700	84.93	46.22	65.15	1.00	1.00	1.00	
800	89.99	59.67	75.22	1.00	1.00	1.00	
900	103.68	69.2	84.25	1.00	1.00	1.00	
1000	106.83	65.42	90.76	1.00	1.00	1.00	

Table 3: Comparison of RT obtained by eDSOS and HGHHM for data instances generated using Left distribution dataset

Task size	eDSOS			HGHHM			T-test
	Max	Min	Avg.	Max	Min	Avg.	
100	18.85	3.17	9.48	1.00	0.90	0.97	t-value= 5.51
200	32.39	11.33	21.15	1.00	0.90	0.99	p-value≈ 0.0003
300	51.26	20.8	35.61	1.00	1.00	1.00	
400	62.14	28.46	48.62	1.00	1.00	1.00	
500	77.17	45.29	66.47	1.00	1.00	1.00	
600	99.89	56.62	76.43	1.00	1.00	1.00	
700	112.46	58.85	91.28	1.00	1.00	1.00	
800	147.41	70.42	108.87	1.00	1.00	1.00	
900	149.57	92.4	121.3	1.00	1.00	1.00	
1000	166.21	93.85	132.59	1.00	1.00	1.00	

Table 4: Comparison of RT obtained by eDSOS and HGHHM for data instances generated using Uniform distribution dataset

Task size	eDSOS			HGHHM			T-test
	Max	Min	Avg.	Max	Min	Avg.	
100	14.93	2.86	9.14	1.00	0.90	0.94	t-value =5.382
200	29.45	8.87	19.9	1.00	0.95	1.00	p-value≈ 0.00044
300	46.86	16.19	31.22	1.00	1.00	1.00	
400	58.46	28.37	45.35	1.00	1.00	1.00	
500	71.22	42.12	56.59	1.00	1.00	1.00	
600	87.61	49.14	69.72	1.00	1.00	1.00	
700	95.6	56	80.95	1.00	1.00	1.00	
800	115.49	61.4	91.51	1.00	1.00	1.00	
900	129.3	78.98	103.7	1.00	1.00	1.00	
1000	134.94	70.92	115.48	1.00	1.00	1.00	

Table Error! No text of specified style in document.: Variation of PIR% based on Response time

DATASETS	Total average RT eDSOS	Total average RT HGHHM	PIR%
Normal	55.05	0.89	98.39
Left-half	48.12	0.98	97.96
Right-half	36.18	0.87	97.60
Uniform	41.49	0.99	97.63

CONCLUSION

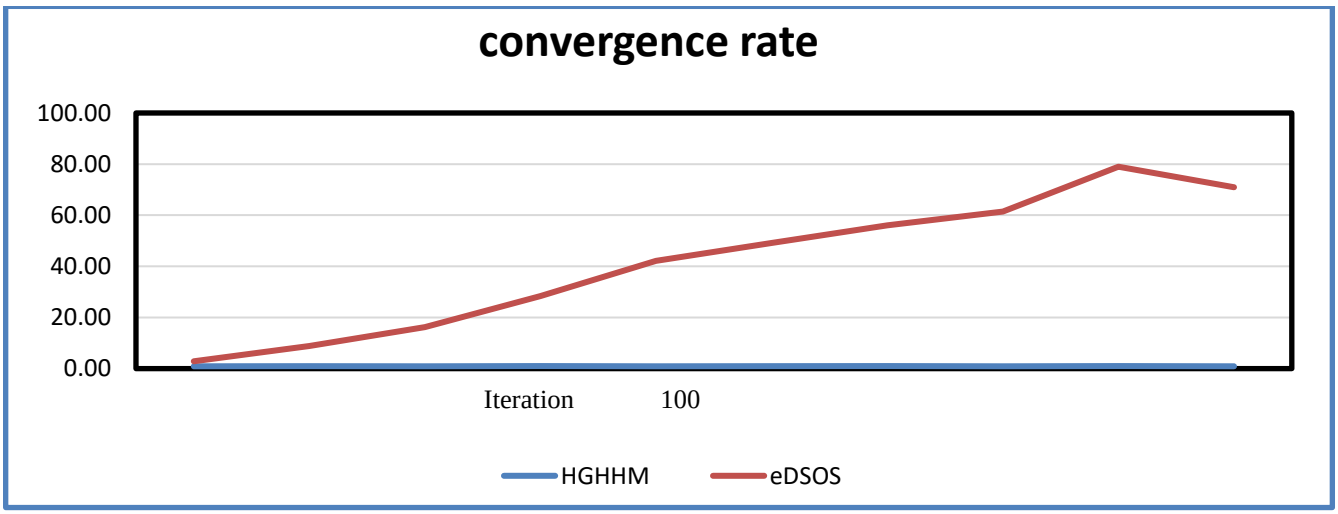


Figure 5: Convergence rate for two algorithms

This study examined how well the Henry Gas–Harris Hawks Modified-Opposition (HGHHM) algorithm would perform for cloud task scheduling. Specifically, the researchers were interested in assessing the HGHHM algorithm's ability to minimize response times, which is a major factor in determining Quality of Service (QoS) and Service Level Agreement (SLA) compliance in cloud-based systems. While prior studies have shown that HGHHM has significantly improved makespan, resource utilization, and degree of imbalance; this study builds on those findings to evaluate the HGHHM algorithm's impact on response times.

CloudSim, an open source cloud computing simulation framework, was used to perform a large number of tests using different types of workload patterns. These tests compared the performance of HGHHM to that of eDSOS (Enhanced Discrete Symbiotic Organisms Search), which served as our baseline. In each test, HGHHM performed better than eDSOS, reducing response times by up to 98 percent and improving resource usage. Paired t-tests were performed to verify the statistical significance of these results. All p-values obtained from the paired t-tests indicated that the differences observed were statistically significant.

HGHHM outperforms other methods because of its balanced use of exploration and exploitation strategies by integrating Harris Hawks Optimization (HHO) and Henry Gas Solubility Optimization (HGSO). The hybrid method improves both local and global searches so that it is able to effectively assign tasks to virtual machines, reduce execution delay, and optimize the utilization of available computing resources.

In summary, the results demonstrate that HGHHM is a flexible, reliable, and adaptable approach for scheduling that can manage multiple and complex jobs in cloud computing. The suggested method is able to meet the many dimensions of scheduling problems in cloud environments by simultaneously decreasing response times and increasing resource usage. Additionally, the decreased response times support the applicability of this method to real world situations where adherence to Quality of Service (QoS) standards and Service Level Agreements (SLAs) is critical.

Future studies could further evaluate the effectiveness and adaptability of HGHHM in different dynamic cloud computing environments. Potential research directions include energy-efficient scheduling to reduce power consumption, fault-tolerant scheduling to improve system reliability, and inter-cloud operations to enhance task allocation across distributed cloud resources. In addition, the algorithm could be evaluated using real-world workload traces and compared with other established scheduling algorithms to provide a broader assessment of its performance. Further investigation into the effects of algorithmic parameters, such as population size and iteration count, could also help improve its optimization performance and computational efficiency. These directions would help assess the practical applicability and scalability of HGHHM in real-world cloud computing environments.

REFERENCES

1. X. Fu, Y. Sun, H. Wang, and H. Li, ‘‘Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm,’’ Cluster Comput., vol. 26, no. 5, pp. 2479–2488, Oct. 2023.
2. S. A. Murad, Z. R. M. Azmi, A. J. M. Muzahid, M. D. K. B. Bhuiyan,M. Saib, N. R. Prottasha, N. J. Prottasha, and A. K. Bairagi, ‘‘SG-PBFS:Shortest gap-priority based fair scheduling technique for job scheduling in cloud environment,’’ Future Generat. Comput. Syst., vol. 150,pp. 232–242, Jan. 2024.
3. N. O. Alkaam, A. B. Sultan, M. Hussin, and K. Y. Sharif, ‘‘The methods used for solving task scheduling problem in cloud computing environment,’’ J. King Saud Univ.-Comput. Inf. Sci., vol. 12, pp. 12538–12544, 2023.
4. A.Abdo, Karamany, T. S., & Yakoub, A. (2024). A hybrid approach to secure and compress data streams in cloud computing environment. Journal of King Saud University - Computer and Information Sciences, 36(3). https://doi.org/10.1016/j.jksuci.2024.101999
5. Alkaam, N. O., Sultan, A. B. M. D., Hussin, M., & Sharif, K. Y. (2024). Hybrid Henry Gas-Harris Hawks modified-opposition algorithm for task scheduling in cloud computing. Journal of Theoretical and Applied Information Technology, 102(15), 5873–5893.
6. Abdulredha, M., Attea, B., & Jabir, A. (2020). Heuristic and Meta-Heuristic Optimization Models for Task Scheduling in Cloud-Fog Systems: A Review. Iraqi Journal for Electrical and Electronic Engineering, 16(2), 103–112. https://doi.org/10.37917/ijeee.16.2.11
7. Houssein, E. H., Gad, A. G., Wazery, Y. M., & Suganthan, P. N. (2021). Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends. Swarm and Evolutionary Computation, 62(January), 100841. https://doi.org/10.1016/j.swevo.2021.100841
8. Dubey, K., & Sharma, S. C. (2021). A hybrid multi-faceted task scheduling algorithm for cloud computing environment. International Journal of Systems Assurance Engineering and Management, https://doi.org/10.1007/s13198-021-01084-0
9. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. Revue d’Intelligence Artificielle, 34(4), 479–485. https://doi.org/10.18280/ria.340413
10. Sharma, P., Shilakari, S., Chourasia, U., Dixit, P., & Pandey, A. (2020). A survey on various types of task scheduling algorithm in cloud computing environment. International Journal of Scientific and Technology Research, 9(1), 1513–1521.
11. Shukri, S. E., Al-Sayyed, R., Hudaib, A., & Mirjalili, S. (2021). Enhanced multi-verse optimizer for task scheduling in cloud computing environments. Expert Systems with Applications, 168(November 2020), 114230. https://doi.org/10.1016/j.eswa.2020.114230
12. Singhal, S., & Sharma, A. (2021). A job scheduling algorithm based on rock hyrax optimization in cloud computing. Computing. https://doi.org/10.1007/s00607-021-00942-w
13. Amer, D. A., Attiya, G., Zeidan, I., & Nasr, A. A. (2022). Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing. Journal of Supercomputing, 78(2), 2793–2818. https://doi.org/10.1007/s11227-021-03977-0
14. Liu, H. (2022). Research on cloud computing adaptive task scheduling based on ant colony algorithm. Optik, 258(February), 168677. https://doi.org/10.1016/j.ijleo.2022.168677
15. Belgacem, A., & Beghdad-Bey, K. (2021). Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost. Cluster Computing, 0123456789. https://doi.org/10.1007/s10586-021-03432-y

16. Xia, X., Qiu, H., Xu, X., & Zhang, Y. (2022). Multi-objective workflow scheduling based on genetic algorithm in cloud environment. Information Sciences, 606, 38–59. https://doi.org/10.1016/j.ins.2022.05.053
17. Sa, S., Muhammed, A., Abdullahi, M., & Abdullah, A. (2021). An Enhanced Discrete Symbiotic Organism Search Algorithm for Optimal Task Scheduling in the Cloud. 1–24
18. Singhal, S., Ali, S., Awasthy, M., Shukla, D. K., & Tiwari, R. (2024). Rock-hyrax: An energy efficient job scheduling using cluster of resources in cloud computing environment. Sustainable Computing: Informatics and Systems, 42, Article 100985. https://doi.org/10.1016/j.suscom.2024.100985
19. Ledmi, A., Ledmi, M., Souidi, M. E. H., Haouassi, H., & Bardou, D. (2024). Optimizing task scheduling in cloud computing using discrete tuna swarm optimization. Ingénierie des Systèmes d'Information, 29(1), [page range]. https://doi.org/10.18280/isi.290132
20. Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments. Journal of Parallel and Distributed Computing, 142, 36–45. https://doi.org/10.1016/j.jpdc.2020.03.022
21. Ebadifard, F., Babamir, S. M., & Barani, S. (2020). A Dynamic Task Scheduling Algorithm Improved by Load Balancing in Cloud Computing. 2020 6th International Conference on Web Research, ICWR 2020, 177–183. https://doi.org/10.1109/ICWR49608.2020.9122287
22. Zhou, Z., Li, F., Zhu, H., Xie, H., Abawajy, J. H., & Chowdhury, M. U. (2020). An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. Neural Computing and Applications, 32, 1531–1541. https://doi.org/10.1007/s00521-019-04119-7
23. Abd Elaziz, M., & Attiya, I. (2021). An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing. In Artificial Intelligence Review (Vol. 54, Issue 5). Springer Netherlands. https://doi.org/10.1007/s10462-020-09933-3