IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 December 2024, accepted 13 January 2025, date of publication 16 January 2025, date of current version 23 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3530860

RESEARCH ARTICLE

Hybrid Henry Gas-Harris Hawks Comprehensive-Opposition Algorithm for Task Scheduling in Cloud Computing

NORA OMRAN ALKAAM[1,2], ABU BAKAR MD SULTAN[2], MASNIDA B. HUSSIN[2], (Senior Member, IEEE), AND KHAIRONI YATIM SHARIF[3]

[1]Iraqi Ministry of Higher Education and Scientific Research, Baghdad 10045, Iraq
[2]Department of Software Engineering and Information System, Universiti Putra Malaysia, Serdang 43400, Malaysia
[3]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Perak 32610, Malaysia

Corresponding authors: Nora Omran Alkaam (nora.omran20@gmail.com) and Abu Bakar Md Sultan (abakar@upm.edu.my)

This work was supported in part by the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia.

ABSTRACT Users can use online data computing services and computational resources from a distance in cloud computing environments. Task scheduling is a crucial part of cloud computing since it necessitates the creation of dependable and effective techniques for allocating tasks to resources. To achieve optimal performance, it requires accurate task allocation to resources. By optimizing task scheduling, cloud computing solutions can decrease processing times, boost efficiency, and improve overall system performance. To address these challenges, this paper proposes an improved version of Henry gas solubility optimization, which is presented as the Henry Gas-Harris Hawks-Comprehensive Opposition (HGHHC) method. This method is based on two elements: comprehensive opposition-based learning (COBL) and Harris Hawks Optimization (HHO). The HHO algorithm was employed as a local search strategy in this suggested algorithm to improve the quality of authorized solutions. Through meticulous analysis of their opposites and selecting an efficient option, COBL improves the less effective options. This method made it easier to improve insufficient solutions, which increased the overall effectiveness of the chosen strategies. The suggested technique was tested using CloudSim on the NASA, HPC2N, and Synthetic datasets. For makespan (MKS), it achieved performance of 34.30, 72.95, and 28.67, respectively. Regarding resource utilization (RU), the corresponding values were 16.92, 28.72, and 25.58. Therefore, the simulated makespan and resource usage of the proposed HGHHC algorithm were better than those of previous approaches. This highlights the effectiveness of hybrid meta-heuristic algorithms in achieving a balance between exploration and exploitation, preventing them from getting stuck in local optima.

INDEX TERMS Cloud computing, Harris hawks optimization, henry gas solubility optimization, task scheduling.

I. INTRODUCTION

The wide adoption of the internet has led to noticeable technological advancements in data processing and storage in recent years. These advancements gave rise to the present cloud computing concept [1], [2], [34], which has revolutionized the way businesses and individuals manage, store, and process data. Cloud computing enables scalable, cost-effective, and instant access to a common pool of configurable computer resources, significantly enhancing flexibility and efficiency. This revolutionary platform allows users to quickly and easily access global data virtually from anywhere at any time. Nevertheless, one of the most pressing challenges in cloud computing is the accurate and reliable assigning of jobs to resources, a critical aspect for optimizing performance and ensuring user satisfaction [3], [4], [35]. The capacity to

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

efficiently and effectively meet all customer requirements is also critical in terms of quality of service (QoS). An efficient job scheduling method can therefore accomplish those objectives in a given amount of time [5], [6]. Various studies have examined different algorithms to potentially address cloud scheduling for instance: heuristic and meta-heuristic algorithms [7], [8].

It's crucial to remember that while heuristic techniques are a useful tool for job scheduling, they don't always provide the best solution. Consequently, meta-heuristic algorithms are considered to be the best approach for handling complicated problems because they are noticeably better than alternative approaches. Instead of using exponential time, these techniques can find roughly optimal solutions in a polynomial amount of time [9], [10].

In order to overcome task scheduling obstacles, this study presents a new hybrid meta-heuristic technique called HGHHC, which is intended to maximize task scheduling in cloud computing by decreasing makespan and improving the efficiency of resource consumption. The proposed technique improves the HGSO algorithm's local search by utilizing comprehensive opposition-based learning and Harris Hawks optimization (HHO). Thus, the main goal of this research was to provide an improved Henry gas solubility optimization method for cloud scheduling. This paper contributes by:

- Propose a robust scheduling algorithm specifically designed for heterogeneous cloud environments.
- Propose the Henry Gas- Harris Hawks- Comprehensive Opposition Algorithm to tackle multi-objective optimization problems, focusing on minimizing makespan while maximizing resource utilization.
- To enrich the literature by presenting a new state-of-the-art sequential hybrid algorithm for job scheduling in cloud computing, offering a valuable reference point for researchers and practitioners in the field.
- Introduce a dynamic scheduling framework that incorporates a rescheduling technique.

This work's next sections are arranged as follows: A thorough summary of important research on the subject is provided in Section II: Related Works. Section III: describes how to formulate the scheduling problem as an optimization challenge. Section IV describes the proposed algorithm. Section V: Experimental Environment: This section provides a thorough evaluation and explanation of the data, as well as a presentation of the experimental findings. The results and analysis are reported in Section VI. Finally, Section VII is the conclusion of our investigation, in which we make concluding observations and recommend potential directions for future research on this topic.

II. RELATED WORK

The current literature has several issues that enable us to create a novel work scheduling algorithm. Even with cloud computing's advances, effectively allocating resources and scheduling work remain difficult jobs. Task scheduling techniques used today in cloud computing environments often face limitations in terms of scalability, resource utilization, and makespan optimization. Many traditional algorithms struggle to maintain performance consistency when workloads vary significantly in size, leading to increased makespan and suboptimal resource allocation, especially as the number of tasks increases. Furthermore, some methods are prone to premature convergence, resulting in local optima and inefficient scheduling solutions. These restrictions highlight the need for creative solutions to deal with these issues and improve the efficiency of cloud computing systems [11], [12].

To begin with the overview, we looked at a variety of meta-heuristic approaches that aim to enhance efficiency by balancing exploration and exploitation strategies. We intensively examined work scheduling studies that employ meta-heuristic algorithms in practice. For example, Fu et al. [13] studied cloud scheduling operations and suggested a hybrid approach that combines particle swarm optimization (PSO) and genetic algorithms (GA). To broaden the search range within the solution space, lowering the probability of the algorithm converging prematurely to a local optimal solution. The method not only decreases makespan but also enhances the accuracy of convergence. The limitation of this method is that it focuses on enhancing only a single objective.

Moreover, Srichandan et al., [14], proposed a hybrid strategy, combining the best aspects of genetic algorithms and bacterial foraging algorithms. The article's key contribution is that the scheduling algorithm minimizes the time required to accomplish a task as well as the amount of energy used. The findings show that the suggested algorithm outperforms competing algorithms with regard to convergence, stability, and solution diversity. The primary limitation of this study is that introducing additional parameters may increase the complexity of the algorithm and potentially impact its overall performance.

Additionally, an improved discrete symbiotic organism search technique combined with meta-heuristics was presented by Sa et al. to maximize job scheduling in cloud computing. Their experiments, conducted using the CloudSim simulator, demonstrated that the proposed approach performed significantly better than the benchmark technique, particularly in large search spaces, with improvements in makespan and response time. However, the method often encountered local optima due to the high values of makespan and response time [26]. Similarly, Singh et al. developed the crow–penguin optimizer [27], a multi-objective approach that optimizes QoS while reducing load and makespan. Despite its benefits, the approach required substantial resources when handling small-sized tasks.

The study of K. Vinoth et al. proposed an optimization strategy aimed at enhancing the efficiency of data centers through effective load balancing. Their approach focuses on distributing workloads evenly across available resources. The method is designed to maintain optimal performance levels, minimize response time, and maximize resource utilization under different operational conditions. This strategy is particularly significant for data centers that handle diverse types of workloads, providing a robust solution that adapts to fluctuations and ensures system stability. However, a noted limitation of their approach is the limited discussion on the algorithm's adaptability and resilience [28].

In the same context, Mangalampalli et al. designed a multi-objective, trust-aware scheduler that prioritizes tasks and virtual machines (VMs) to minimize makespan and energy usage while assigning tasks to the proper virtual resources. To model the job scheduler, the Whale Optimization Algorithm (WOA) was employed and the entire simulation was conducted using CloudSim. Simulation results demonstrated significant improvements in makespan, energy consumption, and total runtime. However, the limitation of their approach is that its performance still requires further improvement [29].

Abd Elaziz and Attiya introduced the HGSWC strategy, combining the Whale Optimization Algorithm (WOA), Henry Gas Solubility Optimization (HGSO), and Comprehensive Opposition-Based Learning (COBL) to optimize task scheduling [15]. Their evaluation showed that HGSWC outperformed benchmark algorithms in makespan (MKS) performance. However, the algorithm's convergence required further refinement, leaving room for improvement in makespan optimization. Building on the insights from these studies, to address the challenges of cloud task scheduling, this research introduces a novel HGHHC algorithm. This approach integrates COBL, HHO, and HGSO operators to optimize performance. Table 1 provides a summary of more related studies.

TABLE 1. Summary of related papers.

Sq	Techniques	Merits	Demerits
31	Executing the Firefly and Genetic algorithms in sequence.	Execution time	Large solution space in Genetic algorithm
32	HHO optimizes task allocation by balancing VM workloads and reducing response times using a PIO-based approach	Enhanced: Makespan response time, computatio n time, cost, load.	With only 500 tasks considered, the results indicate significant room for further improvement
27	Enhanced discrete symbiotic organisms search (eDSOS)	Enhanced: makespan response time	It still becomes trapped in local optima
33	Elite opposition-based learning+ Harris hawks	Minimizin g the duration of the schedule, minimizing the cost of execution, and optimizing the use of resources.	PIR need more improvements
6	Cat swarm optimization and tabu search	Makespan	Deal with small number of users, data size, single objective
34	The whale optimization algorithm is enhanced by utilizing the mutation operator of the bees algorithm.	Minimizes the task completion time and also execution time.	Performance needs to increase
35	Genetic and thermodynami c simulated annealing	Makespan, schedule length ratio, speedup, and efficiency	Incorporating principles from thermodynamics and information theory can improve the current solution by a balance between global and local search.

III. PROBLEM FORMULATION

Our empirical findings emphasize the significant issue of cloud scheduling, which involves efficiently distributing numerous tasks across available computing resources to achieve optimization objectives [12]. The cloud system (CS) is modeled as a collection of N_{pm} physical machines (PMs), each hosting a set of virtual machines VMs. A set of N tasks must be assigned to these VMs, considering the estimated execution time (ETC) matrix, which provides the estimated execution time of each task on each VM. Examine a task that has the index L = 1, 2, 3,..., N, where N is the total number of tasks that have been allocated to virtual machines. The ETC matrix for N jobs and VM virtual machines can be found using the following formula.

$$\mathrm{ETC_{IJ}} = \begin{bmatrix} \mathrm{ETC}_{1,1} & \mathrm{ETC}_{1,2} & \dots & \mathrm{ETC}_{1,\mathrm{VMS}} \\ \mathrm{ETC}_{2,1} & \mathrm{ETC}_{2,2} & \dots & \mathrm{ETC}_{2,\mathrm{VMS}} \\ \dots & \dots & \dots & \dots \\ \mathrm{ETC}_{\mathrm{N},1} & \mathrm{ETC}_{\mathrm{N},2} & \dots & \mathrm{ETC}_{\mathrm{N},\mathrm{VMS}} \end{bmatrix} \tag{1}$$

where the anticipated time Et for L job on the jth VM is represented by the element $\boldsymbol{ETC_{ij}}$, which has the following definition:

$$\mathrm{ETC}_{ij} = \frac{\mathrm{T}\_lenl_i}{MIPS_j} \tag{2}$$

where MIPSj stands for the jth virtual machine's processing power.

The total amount of time needed to finish every task across all of the virtual machines (VMs) that are accessible is known as the makespan. It is effectively the maximum completion time for every VM in the system. Equation (3) provides the formula for makespan, which is:

$$MKS = \max_{j \in 1,2,vms} \sum_{i=1}^{n} \mathrm{ETC}_{i,j} \tag{3}$$

Resource utilization measures how effectively the cloud environment's resources (VMs) are used over the period of time it takes to complete all tasks (i.e., the makespan). It is calculated using equation (4):

$$\text{Resource Utilization} = \frac{\sum_{i=1}^{N} \text{Tvmi}}{\text{makespan} * \text{N}} \tag{4}$$

Tvmi denotes the time required by VMi to complete all assigned tasks, where N represents the total number of resources [11], [16]. Accordingly, the objective is to minimize makespan (MKS) while maximizing resource utilization. The fitness function is defined as follows:

$$Fv = Min\ MKS\ \&\ Fv = max\ RU \tag{5}$$

IV. THE SUGGESTED ALGORITHM

The proposed HGHHC algorithm effectively leveraged the features of the Harris Hawks Optimization (HHO) algorithm to address task scheduling limitations. By functioning as local operators, the HHO components enhanced the performance of the HGSO algorithm. This approach successfully mitigates the limitations of individual meta-heuristic methods. The HHO algorithm was selected due to its four exploitation strategies, which enhance the algorithm's flexibility and effectiveness by providing a balanced approach to exploration and exploitation.

Moreover, HGSO was selected, because of its powerful exploring abilities. The algorithm is kept out of local optima by combining the robust exploration of HGSO with the balanced exploration and exploitation of HHO. This synergy results in a superior algorithm capable of thoroughly exploring the solution space, a critical factor for achieving the objectives of our study.

The proposed HGHHC algorithm begins with generating an initial set of N integer solutions X, sized n to match the number of tasks, with values in the range [1, vms], where vms represents the number of virtual machines. Each solution's fitness value is evaluated using Equation (5), and the best solution, Xb, is identified. HGSO and HHO operators are applied to update solutions based on fitness probabilities, with COBL enhancing the least effective ones. This iterative process continues until termination criteria are met. The algorithm's pseudo-code and structure are detailed in the following sections and illustrated in Fig. 1.

A. FIRST STAGE

The suggested HGHHC algorithm produces solutions represented by Xi, where i = 1, 2, 3,…N. This stage is referred to as the representation stage, the solutions are represented mathematically as follows

$$\begin{aligned} \bar{X}ij &= floor((LBij + \alpha * (UBij - LBij))\ , \alpha \in [0, 1], j \\ \bar{X}ij &= 1, 2, ., vm \end{aligned} \tag{6}$$

According to the task scheduling requirements, the lower bound (LB) is set to 1, while the upper bound (UB) is set according to the number of virtual machines (VMs) available, as indicated in equation (6). In this context, the algorithm employs the floor function to convert real-valued solutions into integer values. This ensures that each Xi is appropriately discretized, aligning with the scheduling constraints and VM assignments required for the cloud computing environment. This integer conversion process is crucial for obtaining feasible task assignments and maintaining the integrity of the scheduling solution.

B. SECOND STAGE

During the update phase, (Fv) is computed for each candidate solution X, providing a measure of its quality and suitability for the optimization process. Based on these fitness values, the best solution, denoted as Xb, is identified as the optimal solution. The algorithm then measures the probability (Pri) of each solution according to its fitness value, as shown in equation (7).

Depending on the probability (Pri) value, the algorithm updates Xi utilizing the Henry Gas Solubility Optimization or Harris Hawks Optimization operators, as described in equation (8). The choice of the operator is influenced by a random value r_{pr} that is generated within the range [1, 0]. This random value r_{pr} is adjusted based on the probability (Pri) as shown in equation (9). Here, Upr Upr and Lpr represent the upper and lower bounds of the probability values, respectively.

The next step is to determine which solutions are the worst, represented by Nw, based on their fitness values, as indicated in equation (10). This process helps in refining the solution set by focusing on improving the least favorable solutions through further updates.

$$Pri = \frac{Fvi}{\sum_{i=1}^{N} Fvi} \tag{7}$$

$$\begin{aligned} &\text{XI}\,(\text{s}+1) \\ &\quad = \left\{ \begin{array}{l} \text{USING OPERATORS OF HHO IF PRI } \geq r_{pr} \\ \text{USING OPERATORS OF HGSO IF PRI } < r_{pr} \end{array} \right\} \end{aligned} \tag{8}$$

$$r_{pr} = Lpr + rnd * (Upr - Lpr) \tag{9}$$

$$Nw = N \times r \times (c_2 - c_1) + c_1 c_1 = 0.1 \text{ and } c_2 = 0.2 \tag{10}$$

C. THIRD STAGE

Subsequently, we applied a technique called comprehensive opposition-based learning (COBL), which makes the algorithm converge on a global solution. The core principle of COBL is to move a solution toward its opposite. We used a tactic as in [17]. Out of the current solutions X and their opposites X^{-}, the best ones were chosen. If the termination criteria were met, the HGHHC algorithm stopped, returning Xb; otherwise, the update process was repeated.

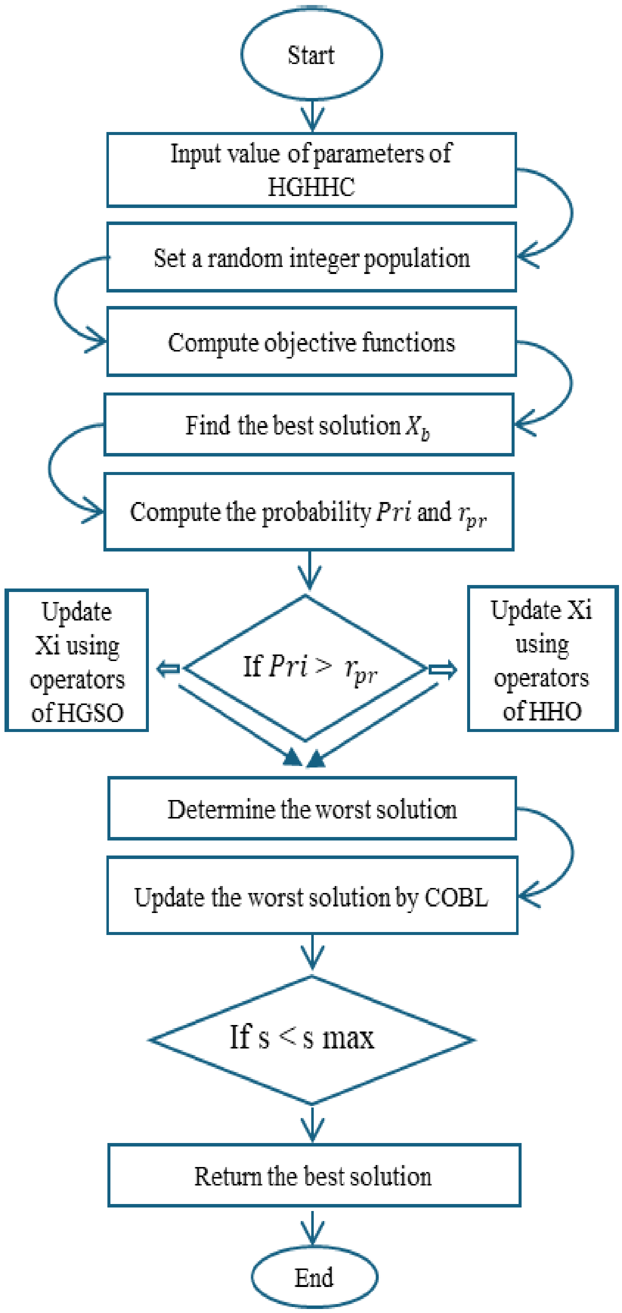


FIGURE 1. The structure of the suggested HGHHC algorithm.

Proposed HGHHC Algorithm pseudo code

1. Assign initial values to the variables of the HGHHC
2. Generate an initial population of N random solutions, each consisting of n elements.
3. For each iteration:
4. Calculate the fitness value of each solution using Equation (5).
5. Determine the solution with the highest fitness value as the best solution (Xb).
6. For each solution:
7. Calculate the probability of exploitation or exploration using Equations (7) and (9).
8. If Pri $\geq r_{pr}$ then
   Update the solution using the exploration phase of HGSO algorithm pseudo code in [25]).
9. Otherwise:
   Update the solution using the exploitation phase of HHO (algorithm pseudo code in [24])
10. Identify the solution with the lowest fitness value as the worst solution.
11. Update the worst solution using (COBL).
12. Increment the iteration counter (s).
13. Repeat the iterative optimization process until the maximum number of iterations (s_max) is reached.
14. Return the best solution (Xb) found during the optimization process.

V. EXPERIMENTAL SETTING

This section details the experimental setup, datasets, the tool used for simulation, and the performance metric employed in this study. We utilized the CloudSim toolkit as our simulation environment. To evaluate the performance of the suggested HGHHC algorithm, we conducted experiments using real-world datasets from HPC2N, NASA, and Synthetic sources. These tasks were assumed to be independent and non-preemptive.

We selected these datasets because they represent a diverse set of real-world and controlled scenarios frequently used in cloud computing research. These datasets are well-known and widely accepted benchmarks that provide authentic and complex workload patterns, allowing us to evaluate the performance of our proposed algorithm under realistic conditions. Moreover, these datasets ensure that our results are comparable with other studies in the field. Additionally, the synthetic dataset allows us to test the algorithm under controlled and customized conditions, ensuring a comprehensive evaluation of its performance in a variety of scenarios.

Each experiment was repeated 30 times to improve the reliability of the results. Furthermore, the performance metrics, including makespan and resource utilization, are discussed in Section III, while PIR is presented next as follows:

The Performance Improvement Rate (PIR) serves as a quantitative measure to evaluate how effectively a proposed methodology surpasses the performance of existing scheduling techniques from prior studies [2]. The PIR is expressed mathematically as follows:

$$PIR = \frac{Zd - Zd'}{Zd} * 100 \tag{11}$$

where Zd′ represents the fitness value of the proposed algorithm, and Zd denotes the fitness value of the algorithm used for comparison [15], [17].

The parameters for the suggested algorithm and the benchmark algorithm that were looked at in this study are listed in Table 2. These HGSWC and HGHHC values were selected based on earlier research [1], [10].

During the implementation phase, HGHHC variables were used, modifications were made, and outcomes were observed. The parameters used in the suggested algorithm were selected in light of previous studies as well as the results obtained.

TABLE 2. Setting parameters.

Algorithm	HGSWC	HGHHC
Parameter	a = 2, b = 1 , l=5E−2,α = 1 , β = 1	a = 1 b = 1 l=5E-2

VI. RESULTS AND DISCUSSION

This subsection provides an in-depth performance evaluation and analysis of the proposed HGHHC algorithm, shedding light on its effectiveness in optimizing two critical metrics: makespan (MKS) and resource utilization (RU). The evaluation process involves a comparative analysis with the existing HGSWC algorithm across three distinct datasets, each varying in size from 500 to 2500 tasks. By employing these datasets, we aim to simulate real-world scenarios of varying workload intensities, ensuring a comprehensive assessment

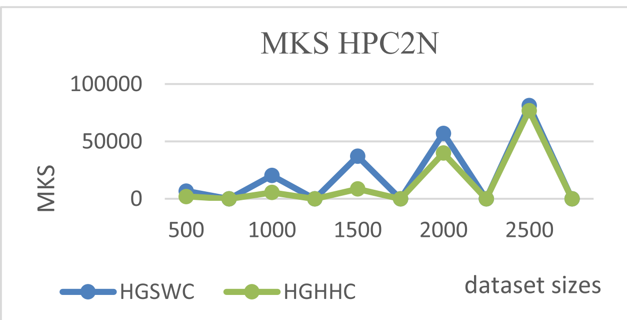


FIGURE 2. Makespan for HPC2N dataset.

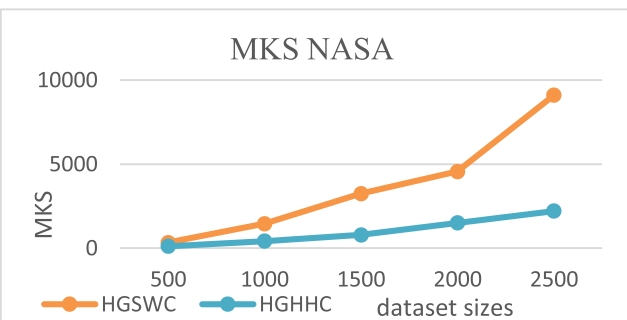


FIGURE 3. Makespan for NASA dataset.

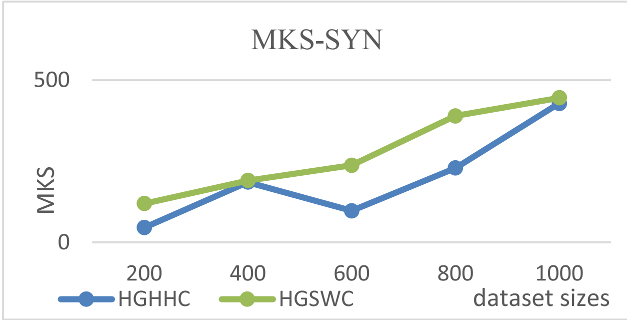


FIGURE 4. Makespan for synthetic dataset.

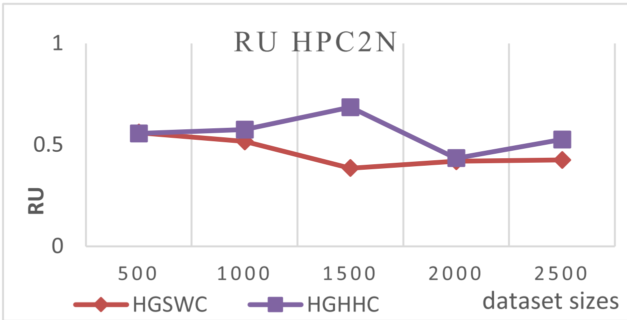


FIGURE 5. Resource utilization for HPC2N dataset.

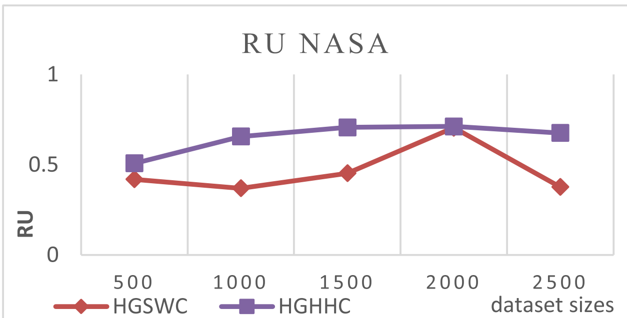


FIGURE 6. Resource utilization for NASA dataset.

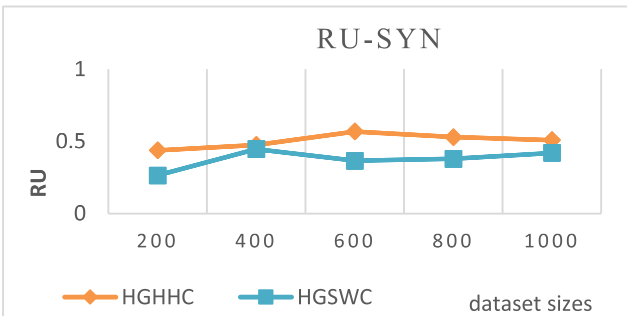


FIGURE 7. Resource utilization for synthetic dataset.

TABLE 3. Best Makespan values for HPC2N dataset.

HPC2N	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
500	6862.57	37698.64	13393.3	2026.071	31894.79	12168.99
1000	20324.27	118289.6	33810.24	5549.968	134387.4	38968.37
1500	37162.76	242464.74	82793.59	8686.952	154059.7	63774.3
2000	57072.06	307797.47	104621.7	39924.88	282161.4	100551.1
2500	81222.63	312714.96	157309.5	76935.63	514296.8	169172.4

TABLE 4. Best Makespan values for NASA dataset.

NASA	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
500	339.5474	2350.236	831.0919133	122.0654	2791.1962	817.3676
1000	1456.596	8565.5688	2968.246471	421.4686	21234.9064	3746.177
1500	3251.124	22489.0262	6954.940207	789.9224	17646.171	6522.258
2000	4569.086	42701.1808	12588.84279	1511.7588	53108.2214	12926.22
2500	9102.564	59202.3444	19898.75494	2216.9566	63527.3898	17924.37

TABLE 5. Best Makespan values for synthetic dataset.

Synthetic	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
200	119.72	784.77	254.06	15.77	93.49	63.53
400	190.64	976.31	355.17	90.56	163.76	115.92
600	237.51	2030.51	625.24	81.74	195.59	159.98
800	389.71	2409.95	836.14	171.19	246.51	209.78
1000	445.25	2400.94	795.09	199.24	349.45	249.47

TABLE 6. Best resource utilization values for HPC2N dataset.

HPC2N	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
500	0.0711222	0.559934	0.23554	0.07864879	0.5555791	0.24678
1000	0.0768226	0.516571	0.26907	0.06980096	0.5742237	0.24601
1500	0.0645975	0.384578	0.19591	0.08236688	0.6845993	0.25344
2000	0.0684602	0.419005	0.24389	0.07687434	0.4336975	0.22606
2500	0.093425	0.424741	0.21587	0.06671982	0.5261247	0.21653

of the algorithm's scalability and robustness. The comparative study offers valuable insights into the capabilities and improvements of the proposed algorithm over the current one.

In our analysis, Fig. 2, 3, and 4 show the best values for MKS for the three datasets. Moreover, Fig. 5, 6, and 7 show the best values for resource utilization. Furthermore, Tables 3, 5, and 4 compare the proposed algorithm to the benchmark algorithm's optimal makespan settings, while Tables 6, 7, and 8 present the RU values. The percentage improvements in makespan (MKS) achieved by the HGHHC algorithm, as shown in Table 9, demonstrate its effectiveness and efficiency compared to benchmark scheduling alternatives across different datasets. Specifically, the algorithm achieved a 34.30% improvement for the HPC2N dataset, a 72.95% improvement for the NASA dataset, and a 28.67% improvement for the synthetic dataset. In addition, the percentage improvements in resource utilization (RU) presented in Table 10 highlight the efficiency of the HGHHC algorithm compared to benchmark scheduling alternatives across various datasets. Specifically, the algorithm

TABLE 7. Best resource utilization values for NASA dataset.

NASA	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
500	0.06408	0.41902	0.205348	0.049599	0.507309	0.204935
1000	0.068164	0.369908	0.212549	0.04	0.656088	0.276226
1500	0.063679	0.452291	0.22442	0.073859	0.706581	0.242423
2000	0.061362	0.704608	0.253378	0.041399	0.712399	0.229277
2500	0.061661	0.376573	0.209399	0.061615	0.675851	0.246432

TABLE 8. Best resource utilization values for synthetic dataset.

Synthetic	HGSWC			HGHHC		
Data size	Min	Max	Avg.	Min	Max	Avg.
500	0.064997	0.263351	0.169812	0.346936	0.644392	0.493882
400	0.052664	0.446511	0.221126	0.174089	0.764089	0.561384
600	0.05966	0.364819	0.198572	0.526576	0.740835	0.637375
800	0.066513	0.378825	0.199576	0.540619	0.809501	0.653232
1000	0.06472	0.419596	0.210318	0.555204	0.796007	0.68528

TABLE 9. Variation of PIR% based on makespan.

DATASETS	Total average makespan (sec) HGHHC	Total average makespan (sec) HGSWC	PIR% improvement over HGSWC
HPC2N	26624.7	40528.85	34.30
NASA	1012.43	3743.78	72.95
SYNTHETIC	197.24	276.56	28.67

TABLE 10. Variation of PIR% based on RU.

DATASETS	Total average RU (sec) HGHHC	Total average RU (sec) HGSWC	PIR% improvement over HGSWC
HPC2N	0.5548	0.4610	16.92
NASA	0.6516	0.4645	28.72
SYNTHETIC	0.5034	0.3746	25.58

achieved improvements of 16.92% for the HPC2N dataset, 28.72% for the NASA dataset, and 25.58% for the synthetic dataset. These results collectively indicate that the HGHHC algorithm provides substantial improvements in both objectives across a diverse range of datasets, showcasing its versatility, scalability, and overall efficiency in cloud environments.

Additionally, we utilized p-values, as detailed in Table 11, based on studies [20], [21], [22], alongside the minimum and maximum values referenced from [15], [18], and [23]. The exceptional performance of the HGHHC algorithm is primarily attributed to its effective balance of exploration and exploitation, which are essential for achieving a balanced and optimized task scheduling process.

The exploration capability allows the algorithm to thoroughly search the solution space and discover diverse task allocation possibilities, reducing the risk of getting trapped in local optima. This ensures that the HGHHC algorithm explores a wider range of potential solutions, leading to more effective resource utilization and scheduling outcomes. Simultaneously, the exploitation capability refines these solutions by focusing on the most promising regions in the solution space, ensuring the algorithm hones in on the optimal configurations efficiently.

TABLE 11. Detail of t-test for MKS.

Description	P- Value
If p-value < 0.05, H_0 reject. This suggests a statistically significant difference between the two algorithms. Conversely, If p-value ≥ 0.05, H_0, fail to reject. Our null hypothesis, H_0, is thus rejected.	<. 01

In summary, the HGHHC algorithm demonstrates superior performance compared to existing algorithms. Which means providing balanced exploitation and exploration to avoid the trapping in local optima.

VII. CONCLUSION AND FUTURE PROSPECTS

This paper introduces the HGHHC method as a novel optimization approach aimed at enhancing the efficiency of a recently meta-heuristic algorithms [15]. The integration of HHO as a local search mechanism improves the exploitation capabilities of HGSO, resulting in superior solution quality. The COBL technique was also used for the worst solutions, which efficiently assigned jobs to the cloud's resources. This algorithm hybridization effectively combines exploitation and exploration abilities. Given that the simulated HGSWC algorithm confirmed the proposed HGHHC algorithm, it outperformed the benchmark algorithm across all test functions.

Consequently, the suggested HGHHC algorithm consistently showed a lower Makespan (MKS) and a higher Resource Utilization (RU) under all test scenarios, proving its optimality. The probability of convergence to local optima is significantly decreased by this balance between exploration and exploitation capabilities. As the number of tasks rises and data environments get more complicated, future research is crucial. Applying the method to more domains, including edge cloud and green cloud. Additionally, further exploration could focus on hybridizing the current algorithm with other meta-heuristic techniques specifically tailored for real-time and dynamic task scheduling in distributed networks.

REFERENCES

[1] A. Belgacem and K. Beghdad-Bey, "Multi-objective workflow scheduling in cloud computing: Trade-off between makespan and cost," *Cluster Comput.*, vol. 25, no. 1, pp. 579–595, Feb. 2022.

[2] 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.

[3] S. E. Shukri, R. Al-Sayyed, A. Hudaib, and S. Mirjalili, "Enhanced multi-verse optimizer for task scheduling in cloud computing environments," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114230.

[4] M. Abdulredha, B. Attea, and A. Jabir, "Heuristic and meta-heuristic optimization models for task scheduling in cloud-fog systems: A review," *Iraqi J. Electr. Electron. Eng.*, vol. 16, no. 2, pp. 103–112, Dec. 2020.

[5] P. Banerjee, S. Roy, A. Sinha, M. M. Hassan, S. Burje, A. Agrawal, A. K. Bairagi, S. Alshathri, and W. El-Shafai, "MTD-DHJS: Makespan-optimized task scheduling algorithm for cloud computing with dynamic computational time prediction," *IEEE Access*, vol. 11, pp. 105578–105618, 2023, doi: 10.1109/ACCESS.2023.3318553.

[6] B. Zakaria, K. Abouelmehdi, A. Beni-Hssane, and H. Khaloufi, "New hybrid algorithm for task scheduling in cloud computing," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 24, pp. 6272–6279, 2021.

[7] B. K. Dewangan, A. Jain, and T. Choudhury, "AP: Hybrid task scheduling algorithm for cloud," *Revue D'Intell. Artificielle*, vol. 34, no. 4, pp. 479–485, Sep. 2020.

[8] M. Tanha, M. Hosseini Shirvani, and A. M. Rahmani, "A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 1–12, Dec. 2021.

[9] M. Gao, Y. Zhu, and J. Sun, "The multi-objective cloud tasks scheduling based on hybrid particle swarm optimization," in *Proc. 8th Int. Conf. Adv. Cloud Big Data (CBD)*, 2020, pp. 1–5.

[10] A. Pradhan, S. K. Bisoy, and A. Das, "A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment," *J. King Saud Univ.–Comput. Inf. Sci.*, vol. 34, no. 8, pp. 4888–4901, Sep. 2022.

[11] M. H. Shirvani and R. N. Talouki, "Bi-objective scheduling algorithm for scientific workflows on cloud computing platform with makespan and monetary cost minimization approach," *Complex Intell. Syst.*, vol. 8, no. 2, pp. 1085–1114, Apr. 2022, doi: 10.1007/s40747-021-00528-1.

[12] K. Sharma, M. M. Hassan, S. K. Pandey, M. Saini, R. Doriya, M. K. Ojha, A. Sinha, C. Kaushal, A. K. Bairagi, and N. F. Soliman, "Cloud based multi-robot task scheduling using PMW algorithm," *IEEE Access*, vol. 11, pp. 146003–146013, 2023, doi: 10.1109/ACCESS.2023.3344459.

[13] 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.

[14] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Comput. Informat. J.*, vol. 3, no. 2, pp. 210–230, Dec. 2018, doi: 10.1016/j.fcij.2018.03.004.

[15] M. Abd Elaziz and I. Attiya, "An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing," *Artif. Intell. Rev.*, vol. 54, no. 5, pp. 3599–3637, Jun. 2021.

[16] M. H. Shirvani, "A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems," *Eng. Appl. Artif. Intell.*, vol. 90, Jan. 2020, Art. no. 103501, doi: 10.1016/j.engappai.2020.103501.

[17] I. Attiya, M. A. Elaziz, and S. Xiong, "Job scheduling in cloud computing using a modified Harris hawks optimization and simulated annealing algorithm," *Comput. Intell. Neurosci.*, vol. 2020, no. 1, 2020, Art. no. 3504642.

[18] 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.

[19] S. Velliangiri, P. Karthikeyan, V. M. A. Xavier, and D. Baswaraj, "Hybrid electro search with genetic algorithm for task scheduling in cloud computing," *Ain Shams Eng. J.*, vol. 12, pp. 631–639, Mar. 2021.

[20] S. Z. M. Jamaludin, N. A. Romli, M. S. M. Kasihmuddin, A. Baharum, M. A. Mansor, and M. F. Marsani, "Novel logic mining incorporating log linear approach," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9011–9027, 2022, doi: 10.1016/j.jksuci.2022.08.026.

[21] N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, S. S. Sidik, A. Alway, N. A. Romli, Y. Guo, and S. Z. M. Jamaludin, "A modified reverse-based analysis logic mining model with weighted random 2 satisfiability logic in discrete Hopfield neural network and multi-objective training of modified niched genetic algorithm," *Expert Syst. Appl.*, vol. 240, Oct. 2024, Art. no. 122307, doi: 10.1016/j.eswa.2023.122307.

[22] M. S. M. Kasihmuddin, S. Z. M. Jamaludin, M. A. Mansor, H. A. Wahab, and S. M. S. Ghadzi, ''Supervised Learning Perspective in Logic Mining,'' *Mathematics*, vol. 10, no. 6, pp. 1–35, 2022, doi: 10.3390/math10060915.

[23] J. Chen, Y. Gao, M. S. M. Kasihmuddin, C. Zheng, N. A. Romli, M. A. Mansor, N. E. Zamri, and C. When, ''MTS-PRO2SAT: hybrid mutation Tabu search algorithm in optimizing probabilistic 2 satisfiability in discrete Hopfield neural network,'' *Mathematics*, vol. 12, no. 5, p. 721, 2024, doi: 10.3390/math12050721.

[24] A. S. Mashaleh, N. F. B. Ibrahim, M. A. Al-Betar, H. M. J. Mustafa, and Q. M. Yaseen, ''Detecting spam email with machine learning optimized with Harris hawks optimizer (HHO) Algorithm,'' *Proc. Comput. Sci.*, vol. 201, pp. 659–664, Jan. 2022, doi: 10.1016/j.procs.2022.03.087.

[25] Y. Chen, L. Gou, and H. Li, ''A differential evolution based Henry gas solubility optimizer for dynamic performance optimization problems of PRO system,'' *Appl. Soft Comput.*, vol. 125, Aug. 2022, Art. no. 109097, doi: 10.1016/j.asoc.2022.109097.

[26] S. Sa'ad, A. Muhammed, M. Abdullahi, A. Abdullah, and F. H. Ayob, ''An enhanced discrete symbiotic organism search algorithm for optimal task scheduling in the cloud,'' *Algorithms*, vol. 14, no. 7, p. 200, 2021.

[27] H. Singh, S. Tyagi, and P. Kumar, ''Crow-penguin optimizer for multiobjective task scheduling strategy in cloud computing,'' *Int. J. Commun. Syst.*, vol. 33, Sep. 2020, Art. no. e4467.

[28] K. V. Kumar and A. Rajesh, ''Multi-objective load balancing in cloud computing: A meta-heuristic approach,'' *Cybern. Syst.*, vol. 54, no. 8, pp. 1466–1493, 2022, doi: 10.1080/01969722.2022.2145656.

[29] S. Mangalampalli, G. R. Karri, and U. Kose, ''Multi objective trust aware task scheduling algorithm in cloud computing using whale optimization,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 2, pp. 791–809, 2023, doi: 10.1016/j.jksuci.2023.01.016.

[30] A. Rajagopalan, D. R. Modale, and R. Senthilkumar, ''Optimal scheduling of tasks in cloud computing using hybrid firefly-genetic algorithm,'' in *Advances in Decision Sciences, Image Processing, Security and Computer Vision*. Cham, Switzerland: Springer, 2020, pp. 678–687.

[31] G. A. P. Princess and A. S. Radhamani, ''A hybrid meta-heuristic for optimal load balancing in cloud computing,'' *J. Grid Comput.*, vol. 19, no. 2, p. 21, Jun. 2021, doi: 10.1007/s10723-021-09560-4.

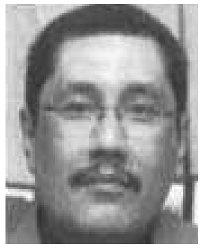
[32] D. A. Amer, G. Attiya, I. Zeidan, and A. A. Nasr, ''Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing,'' *J. Supercomputing*, vol. 78, no. 2, pp. 2793–2818, 2022, doi: 10.1007/s11227-021-03977-0.

[33] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, ''Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment,'' *Comput. Commun.*, vol. 187, pp. 35–44, Jan. 2022, doi: 10.1016/j.comcom.2022.01.016.

[34] H. Hussain, M. Zakarya, A. Ali, A. A. Khan, M. R. C. Qazani, M. Al-Bahri, and M. Haleem, ''Energy efficient real-time tasks scheduling on high-performance edge-computing systems using genetic algorithm,'' *IEEE Access*, vol. 12, pp. 54879–54892, 2024, doi: 10.1109/ACCESS.2024.3388837.

[35] H. Ali, M. S. Qureshi, M. B. Qureshi, A. A. Khan, M. Zakarya, and M. Fayaz, ''An energy and performance aware scheduler for real-time tasks in cloud datacentres,'' *IEEE Access*, vol. 8, pp. 161288–161303, 2020, doi: 10.1109/ACCESS.2020.3020843.

NORA OMRAN ALKAAM was born in Baghdad, Iraq. She received the B.S. and M.S. degrees in computer science from Al-Mustansirya University, Iraq, in 2005 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). Since 2007, she has been with Iraqi Ministry of Higher Education and Scientific Research. Her main research interest includes artificial intelligence. She has published several articles in journals related to this field.



ABU BAKAR MD SULTAN was born in Melaka, Malaysia, in 1965. He received the Ph.D. degree in artificial intelligence from University Putra Malaysia (UPM), in 2007. He is currently the Head of the Information System Department, Faculty of Computer Science, and Information Technology, UPM. His main research interests include artificial intelligence and software engineering, particularly search-based software engineering (SBSE). He has published several articles in conferences and various journals related to SBSE.

MASNIDA B. HUSSIN (Senior Member, IEEE) received the Ph.D. degree from The University of Sydney, Australia, in 2012. She is currently an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM), Malaysia. She is a member of Malaysia Board of Professional Technologies (MBOT), which has the title Technologies (Ts). Her main research interests include the quality of services (QoS), green computing, and resource management for large-scale distributed computing, such as cloud, edge, and fog computing. She has authored or co-authored several research papers published in refereed journals and conference proceedings and also serves on the editorial board for several highly reputation international journals. She has secured several research grants in the areas of computer networks, also in teaching and learning domains. She also actively participates in other research grants with her colleagues.

KHAIRONI YATIM SHARIF received the Ph.D. degree in software engineering from the Lero, the Science Foundation Ireland Research Centre for Software, University of Limerick, Ireland. He is currently an Associate Professor with Universiti Teknologi PETRONAS, Malaysia, and an Adjunct Associate Professor with the Shibaura Institute of Technology, Japan, specializing in software engineering and software security. He has extensive research interests include real-time systems, software security, and programmers' information needs. He has contributed to numerous international research projects and has published extensively in reputable journals. His professional affiliations include IEEE, Malaysia Board of Technologist, and the Psychology in Programming Interest Group (PPIG).