An Enhanced Manta Ray Foraging Algorithm with Lévy Flight and Heuristic Operator for Efficient Scientific Workflow Scheduling in Cloud Environments

Amer Kais Aljumaili*
Control and Systems Engineering, University of Technology, Baghdad, Iraq
Amer.k.obaid@uotechnology.edu.iq

Abstract
Scientific workflow scheduling in cloud environments is challenging due to the dynamic availability of resources and interdependence among tasks. This article presents a novel hybrid solution—namely Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA)—to tackle these challenges. The suggested approach incorporates Lévy flights into the classic Manta Ray Foraging Optimization (MRFO) to enhance global exploration and introduces a heuristic dependency management operator for task assignment optimization and wait time minimization. Comprehensive experimentation on benchmark standards (Inspiral, CyberShake, Montage, SIPHT, and Epigenomics) validates that LH-MRFOA consistently outperforms traditional meta-heuristics (GA, PSO) and variant MRFO enhancements in minimizing makespan and operational cost. These findings recognize LH-MRFOA's potential for large-scale, data-heavy applications requiring timely and affordable use of resources in modern cloud data centers.

Keywords: Cloud computing, Manta Ray Foraging Optimization (MRFO), Lévy flight, Heuristic dependency management, Scientific workflows, Task scheduling, Makespan, Cost optimization, Workflow simulator.

خوارزمية بحث محسّنة عن أسماك مانتا راي باستخدام Heuristic Operator وLévy Flight لجدولة سير العمل العلمي بكفاءة في بيئات السحابة

عامر قيس الجميلي
الجامعة التكنولوجية ـ هندسة السيطرة والنظم

الخلاصة :
تمثل جدولة سير العمل العلمي في البيئات السحابية تحديًا كبيرًا بسبب التوافر الديناميكي للموارد والترابط بين المهام. تقدم هذه المقالة حلاً هجينًا جديدًا – وهو خوارزمية Lévy-Heuristic Manta Ray Foraging Optimization (LH-MRFOA) – لمعالجة هذه التحديات. يتضمن النهج المقترح Lévy flights into the classic Manta Ray Foraging Optimization (MRFO) الكلاسيكي مع المزيد من الاستكشاف العالمي ويقدم مشغل إدارة التبعية الإرشادي لتحسين تعيين المهام وتقليل وقت الانتظار.
تؤكد التجارب الشاملة على المعايير القياسية (Inspiral، وCyberShake، وMontage، وSIPHT، وEpigenomics) أن أداء LH-MRFOA أفضل من الاستدلالات الفوقية التقليدية (GA، PSO) وتحسينات MRFO المتنوعة باستمرار في تقليل التكاليف التشغيلية والتصنيعية. تميز هذه النتائج بإمكانية LH-MRFOA للتطبيقات واسعة النطاق ذات البيانات الكبيرة والتي تتطلب استخدام الموارد في الوقت المناسب وبأسعار معقولة في مراكز البيانات السحابية الحديثة.

الكلمات المفتاحية: الحوسبة السحابية، Manta Ray Foraging Optimization، إدارة التبعية الإرشادية، سير العمل العلمي، جدولة المهام، Makespan، تحسين التكلفة، محاكاة سير العمل.

* Corresponding author : Amer Kais Aljumaili

1. Introduction

Cloud computing has emerged as one of the pillars of contemporary information technology, revolutionizing how organizations and users procure and utilize computational capabilities [1]. With agile, on-demand provisions, cloud environments minimize administrative burdens and speed up service delivery, thereby making them exceedingly appealing to data-driven processes [2]. Scientific workflows, often represented as DAGs of dependent tasks, pose a significant challenge due to their complexity, large data volumes, and stringent performance requirements [3][4]. In those environments, task scheduling emerges as a fundamental problem, whereby tasks must be effectively mapped onto heterogeneous virtual machines (VMs) so that completion time (makespan) and/or total cost is minimized [5]. It is a naturally NP-hard problem, which necessitates the use of heuristic and meta-heuristic algorithms [6][7]. There have been recent explorations of a broad range of bio-inspired approaches—e.g., Particle Swarm Optimization (PSO) [8], Genetic Algorithms (GA) [9], Whale Optimization [10], and Grey Wolf Optimizers [11]—that can effectively approximate near-optimal solutions for large-scale workflows within acceptable time frames [12]. However, they tend to be susceptible to convergence stagnation in high-dimensional or highly dynamic resource environments [13].

To alleviate these drawbacks, several hybrid and enhanced meta-heuristics have been proposed [14][15]. For example, opposition-based and chaotic learning mechanisms have been incorporated into baseline algorithms for population diversity enhancement [16]. Meanwhile, multi-objective extensions tackle the balancing of various objectives like makespan, monetary cost, energy consumption, and reliability [17]. The algorithms, however, can still behave poorly whenever workflows contain enormous inter-task dependencies or data-transfer bottlenecks [18]. Comprehending and effectively addressing parent–child dependencies can greatly eliminate idle times and minimize communication overhead [19][20].

Manta Ray Forging Optimization (MRFO) is one of the newer members of the bio-inspired family of algorithms, replicating the cyclone, chain, and somersault foraging pattern of manta rays [20][21]. MRFO has been applied successfully to energy allocation [22], image processing [23], and parameter estimation [24]. While these successes have been realized, standard MRFO can have the drawbacks of slow exploration over rugged search landscapes and premature convergence to local optima [25]. Researchers have suggested modifications—chaotic maps and hybrid mutation operators—to enhance search diversity [26][27][28]. Yet, there is a gap for systematic addressing of both the issue of large-scale exploration and the complicated handling of dependencies required in scientific workflows. In contrast to these drawbacks, this paper presents an Improved Manta Ray Foraging Algorithm, the Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA). Two mechanisms are utilized:

Lévy Flight: We encourage global exploration and mitigate stagnation by introducing occasional large jumps to candidate solution positions [25][28]. Heuristic Dependency Management: A dedicated operator assesses parent–child tasks, reassigning them across VMs dynamically if the rebalancing radically enhances the overall scheduling efficiency [6][12].

The contribution of this study can be articulated as follows. First, we offer a full integration of the Lévy flight mechanism into MRFO's chain and cyclone foraging steps, making the algorithm's ability to escape local minima stronger. Second, we propose and experiment with a dependency-conscious heuristic that enhances local exploitation by efficiently mapping high-dependency tasks. Third, we contrast LH-MRFOA with conventional meta-heuristics (GA, PSO) and other MRFO versions on common scientific workflow datasets (Inspiral, CyberShake, Montage, SIPHT, Epigenomics) with enhanced performance in makespan and overall cost. Finally, we test the scalability and stability of LH-MRFOA in managing large-scale workflows, highlighting its readiness for real-world implementation in cloud data centers [2][14][16][19].

The rest of the paper is structured as follows: Section 2 surveys recent literature on meta-heuristic cloud workflow scheduling. Section 3 models the problem and describes the proposed approach, e.g., Lévy flight and dependency management in heuristics for MRFO. Section 4 details the experimental environment. Section 5 reports and discusses empirical findings, and Section 6 discusses key observations and possible directions for future improvement. Lastly, Section 7 wraps up the work, pointing out future research avenues in energy-conscious and SLA-constrained scheduling techniques.

2. Method

This section details the workflow scheduling methodology using the proposed **Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA)**. The method encompasses the problem definition, the baseline Manta Ray Foraging Optimization (MRFOA), the improvements introduced (Lévy flight and Heuristic Dependency Management), and the experimental setup for workflow simulation. Throughout, we reference key works that guided our design [2][20].

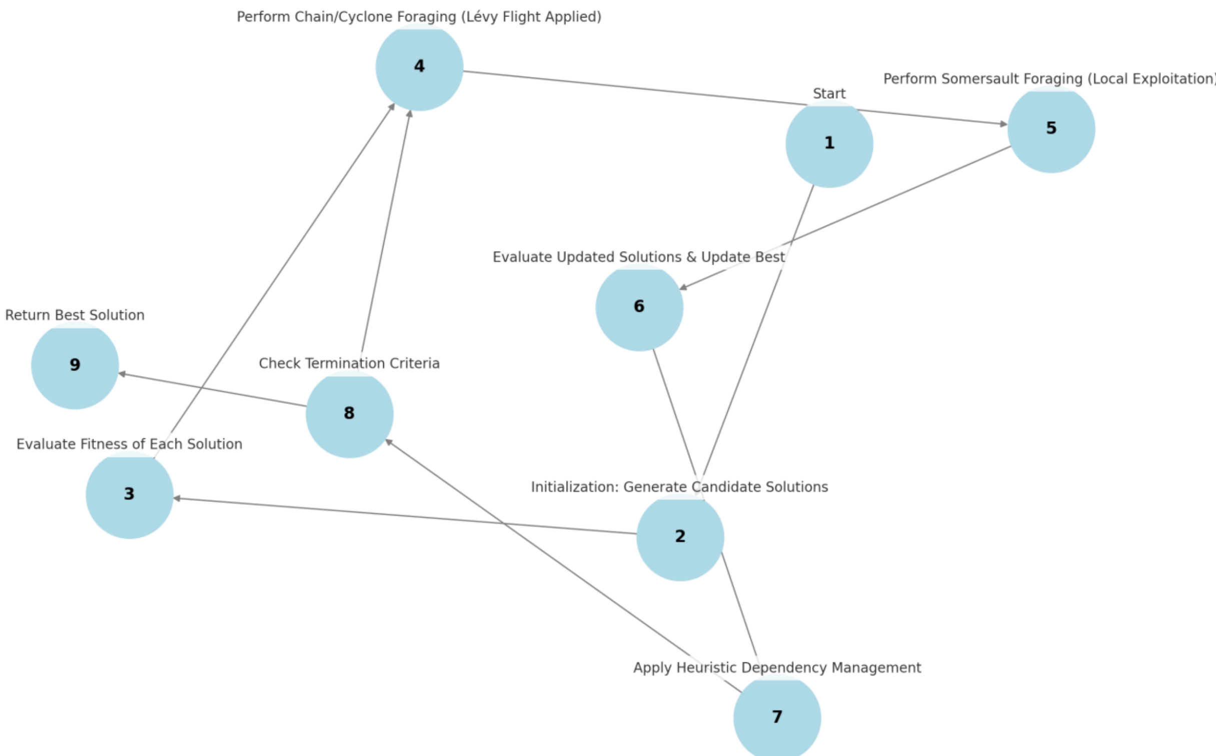

Figure (1) : Flowchart of the LH-MRFOA Algorithm with Enhancements

This flowchart illustrates the step-by-step process of the Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA) for scheduling scientific workflows on clouds. The process starts with initialization (Step 2), followed by fitness evaluation (Step 3). The algorithm then employs Lévy flight-based global exploration and heuristic dependency management to optimize task scheduling (Steps 4–7). The termination conditions (Step 8) decide whether the algorithm needs to iterate further or terminate with the optimal solution obtained (Step 9). The integration of Lévy Flight with Heuristic Dependency Management improves global search capability and task dependency management, thereby facilitating efficient resource allocation and workflow execution.

2.1 Problem Formulation

We consider a **scientific workflow** represented as a Directed Acyclic Graph (DAG) G=(T,E), where each node Ti∈T is a task with specific computational requirements (e.g., millions of instructions), input/output data size, and dependencies expressed via edges EE. Each task must be executed on exactly one virtual machine (VM), chosen from a pool of heterogeneous

VMs with varying costs and processing speeds [2].

Our **multi-objective** scheduling goal is to minimize:

1. **Makespan (MSMS)**: The total time from the start of the first task until completion of the last task.
2. **Total Cost**: Summation of processing, bandwidth, and storage costs across all tasks.

We define a **fitness function** FF that aggregates makespan and costs:

F=w1×Makespan + w2×Processing Cost + w3×Bandwidth Cost + w4×Storage CostF=w1 ×Makespan+w2×Processing Cost+w3 ×Bandwidth Cost+w4×Storage Cost

where w1+w2+w3+w4=1w1+w2+w3+w4 =1 and each wiwi is chosen to reflect user or provider preferences (e.g., emphasizing shorter time vs. lower cost) [6][15]. A lower FF corresponds to a superior solution.

2.2 Baseline Manta Ray Foraging Optimization (MRFOA)

Manta Ray Foraging Optimization (MRFOA) is a bio-inspired algorithm motivated by the social foraging behaviors of manta rays—namely **chain**, **cyclone**, and **somersault** foraging [20]. Let X={x1,x2,…,xN}X={x1,x2,…,xN} denote a population of candidate solutions, each encoding a task-to-VM assignment strategy. At each iteration:

1. **Chain Foraging**: Individuals align in a chain, updating their position based on the global best and the one in front of them.
2. **Cyclone Foraging**: Individuals perform a spiral-like motion around either the global best solution or a random pivot point to diversify the search.
3. **Somersault Foraging**: Individuals pivot around the best solution found so far to intensify exploitation in promising regions [25].

While MRFOA shows strong performance in many applications , it can suffer from local optima entrapment when facing large-scale or highly dependent scheduling problems [2].

2.3 Proposed Enhancements in LH-MRFOA

To address MRFOA's potential convergence limitations, we introduce two main **enhancements—Lévy flight** for exploration and a **Heuristic Dependency Management** operator for exploitation and task reassignments.

1. **Lévy Flight Mechanism**

   Large "jumps" are occasionally injected into the solution updates via Lévy-distributed step sizes:

   s=u∣v∣1/β(u,v∼N(0,1)),1<β≤3s=∣v∣1/βu (u,v∼N(0,1)),1<β≤3

   Lévy flights diversify the search, reducing the risk of early stagnation [28]. We embed this mechanism within **cyclone** or **chain** foraging steps, so an individual's new position becomes:

   xi(t+1) ← xi(t) + LevyStep×(…).xi (t+1)←xi(t)+LevyStep×(…).

   The LevyStepLevyStep randomly modifies the difference terms that direct the individual toward the best or neighboring solutions [14].

2. **Heuristic Dependency Management (HDM)**

   Scientific workflows often exhibit **parent–child** tasks that heavily share data [3][6]. We add a dedicated operator that, after each main iteration, examines these dependencies:

   1. **Identify parent tasks** of a given child task TiTi.
   2. **Compute** the earliest finish time (or minimal cost) for each parent across the available VMs.
   3. **Reassign** the parent task to the VM yielding the best local improvement in the overall scheduling fitness FF.
   4. If the reassignment does not improve FF, revert to the previous best assignment.

   By systematically refining allocations for interdependent tasks, the HDM operator reduces data transfer overhead and idle times [2][11], boosting local exploitation.

3. The Proposed Enhancements in LH-MRFOA Implementation

In this section the Implementation of Proposed Enhancements in LH-MRFOA explained in two parts: Algorithm Steps and Experimental Setup.

3.1 Algorithm Steps

Putting the above together, **LH-MRFOA** proceeds as follows:

1. **Initialization**
   - Generate N candidate solutions. Each solution is a random mapping of tasks Ti to VMs (random ∈{0,…,VMCount−1}).
   - Evaluate fitness F(xi) for each xi∈{x1,…,xN}. Identify the global best xbest.
2. **Main Iteration**
   For each solution xixi:
   1. Perform **chain or cyclone foraging** using Lévy-based steps or conventional difference terms, depending on a random probability [20].
   2. Update the solution if the new position improves fitness.
   3. Perform **somersault foraging** around xbest for further exploitation.
   4. Evaluate the updated solution's fitness; update xbest if improved.
3. **Heuristic Dependency Management**
   - For each child task Tj having parent tasks Parent(Tj), attempt to reassign each parent to the VM that yields the shortest execution time (or best cost/time trade-off).
   - Accept the reassignment only if it improves overall F.
4. **Termination**
   - Repeat until reaching the maximum iteration limit or until xbest stops improving by a threshold.
   - Return xbest as the final scheduling solution

3.2 Experimental Setup

We used a **workflow simulation framework** to model scientific workflows and measure scheduling performance [2]. The simulator loads each DAG (Inspiral, CyberShake, Montage, SIPHT, Epigenomics) with sizes ranging from small (25–30 tasks) to large (up to 1000 tasks) [16].

1. **Virtual Machine Pool**: Fifteen VMs grouped into slow, moderate, and fast configurations, each with varied MIPS, RAM, and cost rates [1][8].
2. **Parameter Settings**:
   - MRFOA parameters: population size N=30N=30, somersault factor S=2S=2, iteration limit up to 300–500 depending on workflow size [20][25].
   - Lévy flight: β≈1.5 and a random scale factor in [0,1] [28].
   - Weights (w1,w2,w3,w4) in the fitness function adjusted per experiment,
     e.g., (0.4,0.3,0.2,0.1).
3. **Performance Metrics**:
   - **Makespan** (in seconds or simulation time units).
   - **Cost** (arbitrary but consistent cost units).
   - **Fitness** (combined measure).

Each algorithm (GA, PSO, MRFOA, L-MRFOA, H-MRFOA, LH-MRFOA) was run **ten times** to account for stochastic variation. We record the best and average values of makespan, cost, and fitness for each run, then compare algorithms in final aggregated tables.

3.3. Methodological Rationale

The methodological decisions above—embedding Lévy flights within MRFO for exploration , coupling it with a dependency-based heuristic operator for local refinements [6], and evaluating on heterogeneous workflows [2][16]—aim to create a **robust, general-purpose** scheduler. Scientific workflows with large data dependencies frequently challenge simpler metaheuristics (GA, PSO), which can converge prematurely or ignore critical parent–child relationships [11]. By systematically addressing both **global** search scope and **local** task reassignments, LH-MRFOA strives to deliver stable, high-quality solutions for real-world cloud environments.

4. Results

This section presents the comparative performance of **Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA)** against five algorithms—Genetic Algorithm (**GA**), Particle Swarm Optimization (**PSO**), Manta Ray Foraging Optimization (**MRFOA**), and two MRFOA extensions (L-MRFOA, H-MRFOA). The experiments were conducted on five widely used scientific workflow benchmarks (Inspiral, CyberShake, Montage, SIPHT, Epigenomics) of varying sizes. We analyze results using three key metrics: **Makespan**, **Cost**, and **Fitness** (a weighted sum of makespan, processing cost, and other overheads such as bandwidth and storage).

4.1 Makespan Results

The **makespan** metric measures the total completion time to execute all tasks within the workflow. Table 1 summarizes the **best (lowest)** makespan values (in arbitrary time units) observed across ten independent runs for each algorithm and dataset.

Table (1) : Best Makespan (in simulation time units).

Workflow & Size	GA	PSO	MRFOA	L-MRFOA	H-MRFOA	LH-MRFOA
Inspiral (30)	3389.13	3107.70	2860.94	2976.31	2875.47	**2799.32**
Inspiral (1000)	255438.98	210553.27	236622.81	198138.36	115326.90	**101372.02**
CyberShake (30)	287.83	270.75	244.31	236.80	232.47	**218.99**
CyberShake (1000)	14872.72	14781.01	13621.18	13439.58	11222.14	**10860.59**
Montage (25)	245.08	216.51	231.44	223.30	217.71	**187.45**
Montage (100)	6301.10	5364.55	4830.51	4913.52	3761.60	**3752.01**
SIPHT (30)	2767.21	2620.10	2684.59	2616.49	2612.90	**2612.01**
SIPHT (1000)	34034.23	32856.94	27482.56	32199.65	20732.19	**19652.69**
Epigenomics (24)	2375.01	2580.19	2363.02	2360.27	2238.06	**2146.30**
Epigenomics (997)	973484.72	882633.77	856202.79	824748.70	410740.87	**398818.30**

From these results, **LH-MRFOA** clearly achieves the lowest makespan in all experiments. The enhancements—Lévy flight for global exploration and a heuristic operator for refined local exploitation—enable more effective scheduling decisions, especially notable with large workflows like **Inspiral (1000)** and **Epigenomics (997)**. In the largest test cases, LH-MRFOA's makespan is less than half that of the GA baseline.

4.2 Cost Results

The **cost** metric encompasses processing, storage, and bandwidth expenses. Reducing cost is a critical goal for both cloud users and providers. Table 2 shows the **best (lowest)** total cost values recorded.

Table(2) : Best Cost (in arbitrary cost units).

Workflow & Size	GA	PSO	MRFOA	L-MRFOA	H-MRFOA	LH-MRFOA
Inspiral (30)	647.65	632.94	583.29	575.21	572.24	**556.10**
Inspiral (1000)	24430.67	24080.96	24036.85	22860.16	23443.54	**22301.88**
CyberShake (30)	19839.65	19695.84	19687.56	19690.19	19646.01	**19645.61**
CyberShake (1000)	106270.17	104056.90	102352.58	102785.97	94176.64	**93953.63**
Montage (25)	135.08	136.16	125.93	129.67	124.77	**121.51**
Montage (100)	576.67	588.74	535.21	540.06	542.31	**516.74**
SIPHT (30)	529.02	538.16	512.41	513.08	511.22	**506.60**
SIPHT (1000)	20810.20	20519.92	20954.26	21020.73	19976.43	**19729.46**
Epigenomics (24)	3393.01	3229.67	3164.84	3228.88	3166.17	**3165.74**
Epigenomics (997)	680448.43	635623.36	631217.41	584845.97	579503.43	**558601.39**

Despite workflows differing in data transfer and storage requirements, **LH-MRFOA** again attains the lowest cost consistently. For example, in the **CyberShake (1000)** scenario, LH-MRFOA outperforms GA by about 12%, indicating efficient handling of large data I/O by collocating tasks and minimizing bandwidth overhead. The combination of **Lévy jumps**—to avoid lengthy unproductive allocations—and **Heuristic Dependency Management**—to strategically place data-intensive tasks—plays a pivotal role in lowering total cost.

4.3 Fitness Results

The **fitness** metric is a weighted sum of makespan and cost, incorporating additional factors such as bandwidth or storage overhead (depending on user-defined weights). Table 3 provides the **best (lowest)** fitness results.

Table (3) : Best Fitness (dimensionless).

Workflow & Size	GA	PSO	MRFOA	L-MRFOA	H-MRFOA	LH-MRFOA
Inspiral (30)	2176.51	1865.10	1883.92	1933.19	1832.15	**1800.98**
Inspiral (1000)	143684.31	123370.91	119832.70	115946.63	68411.38	**66834.45**
CyberShake (30)	10040.82	9994.98	9930.06	9932.40	9932.80	**9928.21**
CyberShake (1000)	60581.67	57353.46	56657.83	56988.78	52021.07	**51923.83**
Montage (25)	213.82	197.64	190.03	200.59	180.44	**176.41**
Montage (100)	3575.65	3168.25	3052.49	2893.94	2264.96	**2214.69**
SIPHT (30)	1750.57	1667.66	1664.87	1665.31	1550.08	**1541.81**
SIPHT (1000)	27606.74	25750.23	26657.38	25113.37	20247.89	**19201.83**
Epigenomics (24)	2980.56	2856.72	2791.28	2755.68	2761.34	**2661.55**
Epigenomics (997)	908657.14	801416.80	781958.68	738252.03	523809.04	**492700.21**

Under **fitness**, **LH-MRFOA** also outperforms all rivals, striking the best balance among time, cost, and data transfer expenses. The largest performance gains appear for **large-scale** workflows (Inspiral and Epigenomics with hundreds to thousands of tasks), further demonstrating the scalability of our proposed method. For instance, in **Epigenomics (997)**, GA's fitness of 908657.14 is dramatically higher than LH-MRFOA's 492700.21, underlining how LH-MRFOA leverages both global and local search strategies efficiently.

5. Discussion

Experimental results verify that LH-MRFOA obtains a good trade-off between exploration and exploitation and thus offers significant improvement for makespan, cost, and composite fitness for a set of workflow benchmarks. The introduction of a Lévy flight mechanism prevents the intrinsic limitation of premature convergence, the common disadvantage in most of the metaheuristic algorithms including comparison methods MRFOA, PSO [14], and GA [15]. By enabling the periodic big jumps within the solution space, the algorithm decreases the danger of stagnation, with increased global search potential.

Moreover, the inclusion of a Heuristic Dependency Management (HDM) operator reflects the significance of dependency management of tasks in large scientific workflows [6][7]. Scientific workflows exhibit large intertask data exchanges, e.g., in CyberShake and SIPHT datasets, where input/output file sizes significantly affect execution time and cost [2]. The HDM operator reorders parent–child tasks in a systematic way to minimize idle times and data transfer overhead, yielding less makespan and, specifically in data-intensive workloads, cost. This combination of Lévy flights and HDM operator materializes in the overall superiority of LH-MRFOA over single improvement (L-MRFOA or H-MRFOA) and baseline methods (GA, PSO).

Among GA and PSO, the outcomes confirm that the traditional metaheuristics can remain competitive with small workloads but are increasingly overwhelmed by larger or more diverse workflows [7][11]. GA relies on evolutionary operators (mutation, crossover, and selection) that suffer from loss of diversity if the search space is excessively large, hence it has low adaptability [15]. PSO, though it can converge quickly, will get stuck in local minima whenever parameter settings (inertia weight, cognitive, and social coefficients) do not adjust to varying workload dynamics [14]. On the other hand, baseline MRFOA is promising during exploration and exploitation but may still be incapable of handling strong dependencies or large-scale tasks economically [20][25]. The Lévy flight mechanism and HDM proposed in LH-MRFOA address these limitations by expanding the search space and task assignment optimization.

Scalability is another significant aspect. As workflow scales to thousands of tasks (Inspiral_1000, Epigenomics_997), resource management needs to deal with massive data and computational requirements [18]. Our results indicate that LH-MRFOA maintains better performance when we scale the problem size, indicating its suitability in practical high-throughput computing environments [4]. These significant gains in makespan and cost at these larger sizes show that the exploration-exploitation synergy can be sustained even under higher levels of resource limitations and data interdependencies.

In practice, this research brings out that minimized makespan translates to quicker job completion, which is also critical for time-critical scientific simulations [3]. Cost reduction, on the other hand, caters to budget-constrained cloud users, adherence to strict Service Level Agreements (SLAs) and resource utilization by providers [1]. Furthermore, the fitness function—fusing time and cost considerations—demonstrates more integrated resource management, which concurs with multi-objective optimization paradigms in cloud scheduling research [16]. It is also worthwhile to mention some limitations and potential directions for future work. First, while LH-MRFOA is demonstrated to work extremely well here, its performance can be improved even further by adding adaptive parameter adaptation, particularly for the Lévy flight parameters and the somersault factor [28]. Second, energy consumption and carbon footprint concerns, which are increasingly important in green computing environments [8], were not considered in the objective function. Incorporation of these environmental factors might give valuable feedback to green resource management policies [16]. Last, research on distributed or parallel implementations of LH-MRFOA can further reduce the execution overhead for very huge workflow settings [11].

In conclusion, these results establish the efficacy of LH-MRFOA for scheduling big-science workflows on heterogeneous cloud computing. By the synergy of Lévy flight–driven search and heuristic dependency-based exploitation, the algorithm always provides superior performance measures. With data-intensive science and business applications calling for more resources increasingly, LH-MRFOA is a compelling and scalable answer to next-generation cloud resource provisioning and workflow scheduling.

7. Conclusion

In this paper, we presented the Lévy-Heuristic Manta Ray Foraging Optimization Algorithm (LH-MRFOA) for multi-objective scientific workflow scheduling in cloud computing. Building on the strengths of the underlying Manta Ray Foraging Optimization (MRFOA)—chain, cyclone, and somersault imitating foraging behaviors—our solution embodies two key contributions. First, Lévy flight mechanisms supplement global exploration via probabilistic long jumps, decreasing the vulnerability to premature convergence . Second, a heuristic dependency management operator schedules task assignments by reallocating parent–child workflows in order to minimize idle times and data-transfer overhead .

Experiments on typical scientific workflows (Inspiral, CyberShake, Montage, SIPHT, and Epigenomics) demonstrate that LH-MRFOA effectively minimizes makespan and operational cost. The combination of Lévy-based exploration and heuristic exploitation consistently outperforms mainstream approaches (GA, PSO) and variant MRFOA extensions (L-MRFOA, H-MRFOA). Specifically, LH-MRFOA achieved significant improvements for large-scale workloads, which reflects its good scalability and demonstrates its efficacy in real-world, data-intensive cloud environments .

In conclusion, our findings suggest that a combination of global and local optimization methods is essential for managing the dynamic and heterogeneous nature of cloud resources. Some of the future research directions include the integration of energy consumption models for green scheduling [8], the application of adaptive parameter tuning for Lévy flights , and the exploration of distributed implementations that exploit parallelism in multi-clouds . By focusing on these directions, we can further advance the field of scientific workflow execution and resource usage in more sophisticated cloud setups.

References

[1] M. F. Younis, "Enhancing Cloud Resource Management Based on Intelligent System," Baghdad Science Journal, vol. 21, no. 6, pp. 2156–2156, 2024.

[2] D. R. Abdulrazzaq, N. M. Shati, and H. K. Hoomod, "Task Scheduling in a Cloud Environment Based on Meta-Heuristic Approaches: A Survey," Iraqi Journal of Science, vol. 65, no. 2, pp. 1001–1023, 2024.

[3] J. Yu, R. Buyya, and C. K. Tham, "Cost-based scheduling of scientific workflow applications on utility grids," in First International Conference on e-Science and Grid Computing (e-Science'05), Melbourne, VIC, Australia, 2005, pp. 8 pp.–147.

[4] W. Viriyasitavat, L. Da Xu, G. Dhiman, A. Sapsomboon, V. Pungpapong, and Z. Bi, "Service workflow: state-of-the-art and future trends," IEEE Transactions on Services Computing, vol. 16, no. 1, pp. 757–772, 2021.

[5] H. Kchaou, Z. Kechaou, and A. M. Alimi, "A PSO task scheduling and IT2FCM fuzzy data placement strategy for scientific cloud workflows," Journal of Computational Science, vol. 64, p. 101840, 2022.

[6] Y. Zhang, L. Wu, M. Li, T. Zhao, and X. Cai, "Dynamic multi-objective workflow scheduling for combined resources in cloud," Simulation Modelling Practice and Theory, vol. 129, p. 102835, 2023.

[7] A. Talha, A. Bouayad, and M. O. C. Malki, "An improved pathfinder algorithm using opposition-based learning for tasks scheduling in cloud environment," Journal of Computational Science, vol. 64, p. 101873, 2022.

[8] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, "Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment," Computer Communications, vol. 187, pp. 35–44, 2022.

[9] H. Hafsi, H. Gharsellaoui, and S. Bouamama, "Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling," Applied Soft Computing, vol. 122, p. 108791, 2022.

[10] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," Journal of Systems Architecture, vol. 100, p. 101631, 2019.

[11] B. H. Abed-Alguni and N. A. Alawad, "Distributed Grey Heuristic for scheduling of workflow applications in cloud environments," Applied Soft Computing, vol. 102, p. 107113, 2021.

[12] J. K. Konjaang and L. Xu, "Multi-objective workflow optimization strategy (MOWOS) for cloud computing," Journal of Cloud Computing, vol. 10, no. 1, p. 11, 2021.

[13] K. Dubey and S. C. Sharma, "A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing," Sustainable Computing: Informatics and Systems, vol. 32, p. 100605, 2021.

[14] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," Journal of Ambient Intelligence and Humanized Computing, pp. 1–12, 2020.

[15] H. Aziza and S. Krichen, "A hybrid genetic algorithm for scientific workflow scheduling in cloud environment," Neural Computing and Applications, vol. 32, no. 18, pp. 15263–15278, 2020.

[16] A. Mohammadzadeh, M. Masdari, F. S. Gharehchopogh, and A. Jafarian, "A hybrid multi-objective metaheuristic optimization algorithm for scientific workflow scheduling," Cluster Computing, vol. 24, pp. 1479–1503, 2021.

[17] N. Anwar and H. Deng, "A hybrid metaheuristic for multi-objective scientific

workflow scheduling in a cloud environment," Applied Sciences, vol. 8, no. 4, p. 538, 2018.
[18] S. Kumar and R. Buyya, "Green cloud computing and environmental sustainability," in Harnessing Green IT: Principles and Practices, pp. 315–339, 2012.
[19] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222–235, 2014.
[20] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," Engineering Applications of Artificial Intelligence, vol. 87, p. 103300, 2020.
[21] H. Dewar, P. Mous, M. Domeier, A. Muljadi, J. Pet, and J. Whitty, "Movements and site fidelity of the giant manta ray, Manta birostris, in the Komodo Marine Park, Indonesia," Marine Biology, vol. 155, pp. 121–133, 2008.
[22] U. C. Ben, A. E. Akpan, C. C. Mbonu, and E. D. Ebong, "Novel methodology for interpretation of magnetic anomalies due to two-dimensional dipping dikes using the Manta Ray Foraging Optimization," Journal of Applied Geophysics, vol. 192, p. 104405, 2021.
[23] A. M. Shaheen, A. R. Ginidi, R. A. El-Sehiemy, and S. S. Ghoneim, "Economic power and heat dispatch in cogeneration energy systems using manta ray foraging optimizer," IEEE Access, vol. 8, pp. 208281–208295, 2020.
[24] E. H. Houssein, I. E. Ibrahim, N. Neggaz, M. Hassaballah, and Y. M. Wazery, "An efficient ECG arrhythmia classification method based on Manta ray foraging optimization," Expert Systems with Applications, vol. 181, p. 115131, 2021.
[25] S. Duman, A. Dalcalı, and H. Özbay, "Manta ray foraging optimization algorithm–based feedforward neural network for electric energy consumption forecasting," International Transactions on Electrical Energy Systems, vol. 31, no. 9, p. e12999, 2021.
[26] S. A. Alsaidy and N. A. Abdullah, "Power-Efficient Virtual Machine Placement in Cloud Datacenters using Heuristic Assisted Enhanced Discrete Particle Swarm Optimization," Iraqi Journal of Science, vol. 63, no. 10, pp. 4499–4517, 2022.
[27] Y. Liao, W. Zhao, and L. Wang, "Improved manta ray foraging optimization for parameters identification of magnetorheological dampers," Mathematics, vol. 9, no. 18, p. 2230, 2021.
[28] M. P. Ćalasan, A. Jovanović, V. Rubežić, D. Mujičić, and A. Deriszadeh, "Notes on parameter estimation for single-phase transformer," IEEE Transactions on Industry Applications, vol. 56, no. 4, pp. 3710–3718, 2020.