Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.050406

ARTICLE

A Novel Optimization Approach for Energy-Efficient Multiple Workflow Scheduling in Cloud Environment

Ambika Aggarwal[1], Sunil Kumar[2,3], Ashok Bhansali[4], Deema Mohammed Alsekait[5,*] and Diaa Salama AbdElminaam[6,7,8]

[1]School of Computer Science, University of Petroleum and Energy Sciences, Dehradun, Uttarakhand, 248007, India

[2]Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, 140401, India

[3]Department of Computer Science, Graphic Era Hill University, Dehradun, 248001, India

[4]Deptartment of Computer Engineering and Applications, GLA University, Mathura, 281406, India

[5]Department of Computer Science and Information Technology, Applied College, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[6]Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha, 13511, Egypt

[7]MEU Research Unit, Middle East University, Amman, 11831, Jordan

[8]Faculty of Computers Science, Misr International University, Cairo, 11800, Egypt

*Corresponding Author: Deema Mohammed Alsekait. Email: Dmalsekait@pnu.edu.sa

Received: 06 February 2024 Accepted: 18 April 2024 Published: 17 July 2024

ABSTRACT

Existing multiple workflow scheduling techniques focus on traditional Quality of Service (QoS) parameters such as cost, deadline, and makespan to find optimal solutions by consuming a large amount of electrical energy. Higher energy consumption decreases system efficiency, increases operational cost, and generates more carbon footprint. These major problems can lead to several problems, such as economic strain, environmental degradation, resource depletion, energy dependence, health impacts, etc. In a cloud computing environment, scheduling multiple workflows is critical in developing a strategy for energy optimization, which is an NP-hard problem. This paper proposes a novel, bi-phase Energy-Efficient Fruit Fly-based Optimization (E^2FFO) algorithm for optimizing energy consumption for scheduling multiple workflows. In the first phase, the proposed E^2FFO algorithm uses first come, first serve, priority scheduling and a Genetic Algorithm to generate the initial workflow search space. In the second phase, the energy consumption is optimized by the proposed E^2FFO algorithm. Eight NAS benchmarks and five NAS classes (A, B, C, S & W) are employed as a case study. The simulation results are carried out on the WorkflowSim 1.0 platform to test the efficacy of the proposed E^2FFO algorithm. The experimental results are compared against energy-aware for workflow scheduling and virtual machine consolidation (EASVMC), Power-Efficient Scheduling for Virtual Machine Systems (PESVMS), Energy Efficiency Scheduler (EES), and heterogeneous earliest finish time (HEFT) algorithms and outperformed them with 10.518%, 16.302%, 26.154%, and 28.982%, respectively, based on average energy consumption on five scientific workflows comprised Montage, CyberShake, Laser Interferometer Gravitational-Wave Observatory (LIGO), Scripps Institution of Oceanography High-Throughput (SIPHT), and Epigenomics.

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

KEYWORDS
Cloud computing; fruit-fly optimization; energy consumption; scientific workflow; NAS benchmarks

1 Introduction

Cloud computing uses the pay-per-use model for sharing virtual resources to run large-scale business/scientific applications by providing high-performance computing, scaling, and virtualized resources [1]. This process is known as task/workflow scheduling. Due to insignificant scheduling techniques in cloud computing, the multiple workflow scheduling problems are divided into several single workflow scheduling problems [2]. Scheduling multiple workflows is essential to execute large-scale applications, especially in data centers with higher electricity consumption. Some of the world's largest data centers containing thousands of IT devices have witnessed exponential growth in internet users and increased energy consumption [3,4]. According to [5], in a data center, direct electricity consumption by servers is 45%, by cooling systems 41%, 11% by storage devices, and the remaining 3% by network and sharing devices. As a result, the CPU consumption lies between 35%–50%, so the power usage in data centers is floundered. Saving maximum energy leads to minimizing the overall budget and reducing carbon emissions, enhancing system reliability, and reducing the cooling cost of data centers [6,7]. In cloud computing environments, workflow scheduling is considered an NP-hard problem [8,9]. Since the last decade, various workflow scheduling techniques have been proposed that focus on optimizing multiple objectives, such as cost, makespan, and deadline, without bothering about energy utilization, that proficiently handle large-scale business applications [10].

For a fixed number of heterogeneous processors, critical path on a processor (CPOP) [11] and heterogeneous earliest finish time (HEFT) [12] scheduling algorithms were proposed to get the optimum scheduling time and high performance. The HEFT picked the highest value of the upward-ranking task at each level. It allocated tasks to the processor, obtaining minimized finish time whereas to prioritize the tasks, the summation values of the downward and upward ranking were used by CPOP. In [13], Bahnasawy et al. discussed sorted nodes in leveled DAG division (SNLDD) for heterogeneous distributed computing systems (HeDCS). SNLDD divided the directed acyclic graph into sublevels and, as per the computation size of jobs scheduled the sorted jobs (in descending order) in each sublevel. A comparative study presented that SNLDD outperformed the Longest Dynamic Critical Path (LDCP) algorithm for parameters like quality of system behavior, schedule length, efficiency, and speed [14]. The SNLDD worked on HeDCS for static task scheduling only. In [15], an efficient workflow-scheduling algorithm (EWSA) was proposed to calculate the execution time for dynamically assigned tasks. To execute the tasks within the deadline, the EWSA algorithm created an apt virtual machine (VM) for the entire application with the minimum resources. To evaluate the performance of EWSA, different workloads for a single workflow can be taken to gain better results. Reference [16] used a genetic algorithm (GA) based approach to find the appropriate solutions for load balancing, makespan, and cost on the Montage workflow. The authors identified an excellent solution based on the best solution for each parameter. The algorithm outperformed standard GA [17], specialized scheduler model, and particle swarm optimization (PSO) [18]. In [19], a hybrid GA and PSO is proposed to schedule workflow on the heterogonous cloud and reduce cost, makespan, and load balancing. In the first phase, GA techniques generate the initial population for scheduling (previously, priority-based FCFS techniques were used). PSO receives the newly generated population and finds

the optimal solution for defined parameters. The presented work was limited to one data center containing a homogeneous environment. In [20], a hybrid Gravitational Search Algorithm (GSA) and HEFT-based technique were proposed to optimize cost and makespan workflow scheduling. A new parameter, cost time equivalence, was introduced to make optimization more faithful. For performance metrics, monetary cost ratio (MCR) and schedule length ratio (SLR) were considered to calculate the efficiency of the proposed technique. This work is limited to the fixed bandwidth between virtual machines (VM) and a single workflow used to evaluate the efficiency of the presented algorithm. Reference [21] proposed a fuzzy dominance sort-based heterogeneous earliest-finish-time (FDHEFT) technique. FDHEFT used a sorting mechanism based on fuzzy dominance and HEFT scheduling heuristic. The proposed algorithm claimed higher cost-makespan tradeoff fronts with better speed than the existing algorithms. The presented approach does not consider scenarios like where makespan and cost overhead the storage and communication time.

We proposed a novel Energy-Efficient Fruit Fly-based Optimization (E^2FFO) algorithm for workflow scheduling to optimize overall energy consumption in a cloud data center. This research presents an enhancement in the fruit fly optimization algorithm for better energy utilization without too much negotiation in traditional objectives. The major contributions are summarized as follows:

- E^2FFO uses first come, first serve, priority scheduling and genetic algorithm to generate the initial workflow search space. Amalgamate three algorithms with the proposed and found that GA showed better alignment.
- The proposed E^2FFO algorithm optimizes energy consumption with eight NAS benchmarks and five NAS classes (A, B, C, S & W).
- The performance evaluation and comparative study show that the proposed E^2FFO algorithm outperformed EASVMC, PESVMS, EES, and HEFT algorithms with 10.518%, 16.302%, 26.154%, and 28.982%, respectively, based on average energy consumption on five scientific workflows comprises Montage, CyberShake, LIGO, SIPHT, and Epigenomics.

The organization of the paper comprises the following sections. Section 2 demonstrates the background details related to nature-inspired algorithms for energy consumption. Section 3 highlights the problem formulation of the energy model for workflow scheduling. Section 4 is the proposed E^2FFO algorithm. Section 5 shows the results and discussions on performance evaluation and comparative analysis. Section 6 gives a brief conclusion with future directions.

2 Related Work

This section discusses the related information and literature survey on energy-efficient-based multiple workflow scheduling techniques. The multiple workflow scheduling problems have been studied through existing literature work. The researchers mainly targeted minimizing the budget, reducing makespan, and executing tasks within the deadline to ensure the QoS in workflow scheduling. While achieving these objectives, it is still a crucial point for researchers to use energy resources efficiently. The literature study showed that few techniques use the VM, VM consolidation, dynamic voltage, frequency scaling (DVFS), and task migration scheme to reduce overall energy consumption. VM levels, poor knowledge, and physical infrastructure constraints presented more challenges to this [22].

The following papers considered energy consumption as a main or secondary objective that needs to be minimized. In [23], Durillo et al. proposed an energy-efficient list-based technique, considering energy utilization and performance for multi-objective workflow scheduling. Cao et al. [24] developed

an algorithm based on the DVFS scheme to improve energy efficiency in scientific workflows of data centers. The proposed algorithm took four server modes, i.e., idle, active, transition, and sleep. In [25], Bousselmi et al. reduced the data communication between workflows by partitioning methods to minimize energy consumption. Then, they applied the cat swarm optimization technique for scheduling partitions on VMs to reduce overall network energy intake. In [26], Chen et al. proposed an energy-efficient online scheduling (EONS) algorithm to schedule various tasks from heterogeneous workflows. EONS dynamically adjusted the system resources to maintain the host's frequency.

Rehman et al. [27] recently designed a Multi-Objective Genetic Algorithm (MOGA) to optimize lifespan, cost, deadline, and energy consumption using DVFS. A gap search algorithm was used for cloud resource optimization. Garg et al. [28] have proposed a reliable and energy-efficient workflow-scheduling algorithm. This algorithm executes in four sub-modules: The priority calculation module, tasks clustering, target time distribution, and cluster assigning with suitable frequency/voltage levels. In [29], Ahmad et al. proposed a scheduling algorithm called reducing energy consumption using fair pre-assignment of available budget (RECFPAB). The proposed algorithm optimizes the energy consumption within the client budget by reducing the schedule length using different budget-constrained scheduling methods. Hussain et al. [30] proposed an Energy and Performance-Efficient Task Scheduling (EPETS) technique for the heterogeneous virtualized cloud. EPETS works in two phases: The initial phase helps to minimize the execution time without bothering about the energy consumption, and the next phase optimizes the energy consumption by discovering the best execution place of the scheduled task within the deadline constraint. The task priority scheme is used for an efficient energy system that balances energy saving and task scheduling. Medara et al. [31] proposed an energy-aware for workflow scheduling and virtual machine consolidation (EASVMC) approach based on water wave optimization for resource utilization and VM migrations. Mohanapriya et al. [32] proposed a PESVMC algorithm that merged the VM scheduling and VM consolidation problem to schedule the workflow tasks. Huang et al. [33] proposed EES to minimize energy consumption rate compared to performance-based SLA. The following paragraph discusses the problems and gaps identified in these works.

While scheduling multiple workflows, researchers have mainly focused on makespan, load balancing, deadline constraints, budget constraints, storage, bandwidth, memory requirement, and QoS Support. Unfortunately, energy consumption is a less targeted parameter, and few techniques have been developed to optimize energy usage in a cloud data center during workflow scheduling [34]. Further, many research efforts [35–37] have been done to reduce/optimize energy consumption using DVFS, adaptive multi-objective task scheduling (AMTS), and E-PAGA techniques by decreasing operational voltage and frequency of the processor leading to the degradation in response time. Still, their scopes are normally limited to the particular resource site or within the homogeneous cluster. Nevertheless, optimization of energy consumption at a complex cloud data center is still to be explored.

3 Problem Formulation

3.1 Energy Model for Workflow

In data centers, the energy consumption of servers is due to memory devices, network devices, CPUs, storage devices, and other extensively used circuits. Among these devices, the CPU consumes the maximum amount of energy. Traditional energy monitoring techniques correlate energy consumption and CPU utilization [38]. A physical server with zero loads consumes 50%–70% energy compared to a physical server running with maximum load capacity [39]. Therefore, the total power utilization by a server can be obtained by Eq. (1).

$$P_{tu} = P_{CPU_i} + (P_{CPU_f} - P_{CPU_i}) * CPU_{tu} \tag{1}$$

where P_{tu} is total power utilization, P_{CPU_i} does idle CPU consume the power, and P_{CPU_f} does the maximum loaded CPU consume the power and CPU_{tu} is the total CPU utilization. Energy consumption (E_c) obtained when the machine is running is computed by Eq. (2).

$$E_c = P_{tu} * Time \tag{2}$$

The energy utilization of task t_k executing on the virtual machine V_m is represented using Eq. (3).

$$E(t_k) = P_{V_m} * Time(t_k, V_m) \tag{3}$$

where P_{V_m} is the power consumption of V_m and $Time(t_k, V_m)$ is the running time of a task t_k on V_m. The total energy consumption for an executing workflow is given by

$$E(W_f) = \sum_{k=1}^{K} E_{t_k} \tag{4}$$

3.2 Problem Definition (Objective Function)

As per Eq. (4), the minimum energy consumption model for heterogeneous workflows is required. The objective function for an efficient energy optimization model is represented by Eq. (5).

$$f(Obj) = \underset{\substack{x = 1, 2, 3, \ldots N \\ y = 1, 2, 3, \ldots .F}}{^{Maximum}} \left(\overbrace{E(W_f).\xi_{xy}}^{Minimize} \right) \tag{5}$$

where x represents the number of nodes and y is the related available frequency of each node in heterogeneous workflows. Then, the objective function $f(Obj)$ can be used for any workflow energy model with scaling factor ξ that can help to choose the optimal solution.

4 Proposed Algorithm

This section presents a novel E²FFO algorithm for scheduling multiple workflows for data center applications.

Algorithm 1: E²FFO (Energy-Efficient Fruit Fly-based Optimization) for workflow scheduling

Input:

Q	*Swarm size* (Set of scheduled workflows)
SL_l	*Initial location of individual swarm* $\forall\ 1 \in \{1, 2, 3, \ldots, L\}$
Max	*Maximum no of iteration, i.e.,* {*20–40*}
S_{pow_α}	*Static powers of each swarm*
D_{pow_α}	*Dynamic powers of each swarm*
CM_{W_f}	*During each iteration, the communication time of all tasks with the highest frequency in a single workflow*
CP_{W_f}	*During each iteration, the computation time of all tasks with the highest frequency in a single workflow*
Fq_{max}	*Maximum frequency of all tasks*
Fq_{d_g}	*Differences between the two frequencies of each task*

(Continued)

Algorithm 1 (continued)

Output: Pareto optimal solution

$$\max_{S_N \in [LB_N, UB_N]_{N=1,2,3...,K}} f(P,E) = \{S_1, S_2, S_3 \ldots, S_N\}$$

$$\therefore QoS = P - E \quad \text{and } Out_{max} = \max(QoS)$$

1. $f(Obj) = \overset{Maximum}{\underset{y=1,2,3,\ldots,F}{x=1,2,3,\ldots,N}} \left(\overbrace{P(W_f).\xi_{xy}}^{Maximize} - \overbrace{E(W_f).\xi_{xy}}^{Minimize} \right)$
2. for Max ← 1 to r do
3. $x_r^{EF} = x_\omega + RV, y_r^{EF} = y_\omega + RV$
   // $\left(x_r^{EF}, y_r^{EF}\right)$ *the initial position of each swarm particle, RV = (0, 1)*
4. *Calculate* $Dist_r = \sqrt{\left(x_r^{EF}\right)^2 + \left(y_r^{EF}\right)^2}$

$$S_r^C = \frac{1}{Dist_r}$$

// $Dist_t$ is the distance between individual fruit fly and food and S_t^C is smell concentration

5. $Smell_r = f\left(S_r^C\right)$ *//for each fruit fly*
6. $\overline{F}(Smell_r) = \frac{1}{v} \sum_{r=1}^{v} f_r(Smell_r)$
7. Call **Algorithm 2**
8. Update swarm particles' location with energy $(x_\omega, y_\omega, z_\omega)\{\{\{$
   - 8.1 $x_\omega = x_\omega + x_\omega * RV(0,1) + x_\omega * \overline{F}(Smell_r)\}$
   - 8.2 $y_\omega = y_\omega + y_\omega * RV(0,1) + y_\omega * \overline{F}(Smell_r)\}$
   - 8.3 $z_\omega = E\left(W_f\right)\xi_{xy}\}$
   - 8.4 Go to step 3.
9. **End for**

The proposed E²FFO algorithm optimizes the previous solution and forms a new solution using the smell concentration function. All input and output parameters are well explained in Algorithm 1. The objective function is used to maximize the difference between performance and energy. Here, f(P, E) = {S1, . . . , SN} is a set initial workflow schedule. Max is used for the maximum number of iterations. The GA approach used for the initial position of each swam particle (individual solution) $\left(x_r^{EF}, y_r^{EF}\right)$. Calculate the mean of smell concentration. $\overline{F}(Smell_t)$ and update the new swarm position.

Algorithm 2: Energy consumption and CPU performance with frequency scaling (ξ_g)

1. $CP_{Scale_g} = \frac{\overbrace{g = \{1,2,3,\ldots\ldots,G\}}^{\max}\left(CP_{Time_g}\right)}{CP_{Time_g}}$
2. $Fq_g = \frac{Fq_{max_g}}{CP_{Scale_g}} \qquad \forall g \in \{1,2,$
3. $Time_{pre} = \overbrace{g = \{1,2,3,\ldots\ldots,G\}}^{max}(CP_{Time_g} + CM_{Time_g})$

(Continued)

Algorithm 2 (continued)

4. $E_{total} = \sum_{g=1}^{G}(D_{pow_g} * CP_{Time_g} + S_{pow_g} * Time_{pre})$
5. ***while*** $(\forall g \neq Fq_{min})$ ***do***
6. ***if*** $(Fq \neq Fq_{last})$
   a. $Fq_g = Fq_g - Fq_{d_g}$
   b. $\xi_g = \frac{Fq_{max_g}}{Fq_{g_{max}}}$ ***endif***
7. $Time_{new} = \overbrace{g = \{1, 2, 3, \ldots, G\}}(CP_{Time_g} * \xi_g) + \min(CM_{Time_g})$
8. $E_{red} = \sum_{g=1}^{G}(S_g^{-2} D_{pow_g} * CP_{Time_g} + S_{pow_g} * Time_{new})$
9. $P\left(W_f\right) = \frac{Time_{pre}}{Time_{new}}$ *and* $E\left(W_f\right) = \frac{E_{red}}{E_{total}}$
10. $QoS = P\left(W_f\right) - E\left(W_f\right)$ ***endwhile***

Algorithm 2 shows the energy consumption and found CPU performance with frequency scaling (ξ_g). The first two steps find the frequency of nodes that consume more energy. Steps 3 and 4 calculate the total time and energy of each node. Step 5 is used for the maximum number of predefined iterations. The next step calculates the frequency scaling factor. Steps 7 and 8 calculate the new time and amount of reduced energy. Step 9 evaluates the performance and total amount of saved energy. The overall complexity of the proposed work is $O(n^2)$.

5 Performance Evaluation

5.1 Experimental Setup

The simulation experiments were implemented using the proposed E²FFO algorithm on the workflows im 1.0 Toolkit [40]. This toolkit supported additional features for workflow management and analysis of different workflow scheduling parameters. Table 1 shows various simulation parameters used for simulation setup.

Table 1: Simulation setup

Hardware	CPU	Intel® Core™ i7-10710U CPU @ 4.70 × GHz 1.10 GHz, 6 core
Software	RAM	64 GB
	External storage	2 TB
	Internet	Gigabit ethernet
	Operating system	Windows 10 Pro
	JAVA	Java version "16.0.1" 2021-04-20
		Java (TM) SE Runtime Environment (build 16.0.1 + 9–24)
		Java HotSpot (TM) 64-Bit Server VM (build 16.0.1 + 9–24, mixed mode, sharing)
	Middleware	Eclipse
	Compiler environment	WorkflowSim-1.0

5.2 Evaluation Parameters

For simulation purposes, five heterogeneous workflows that are LIGO, SIPHT, Montage, CyberShake, and Epigenomics, are used with the graph properties like heterogeneity factor (HF), scalability, communication to computation time ratio (CCTR), range of computation cost with three different power consumption scenarios. These five workflows are considered with eleven different workloads mentioned in Table 2. As per the nature of such workflows, the initial workload for each workflow was taken, and almost the same workload assumptions were considered in [41,42]. The initial workloads are 20, 40, 70, 90, and 100 of Epigenomics, Montage, LIGO, CyberShake, and SIPHT. To compare better, all workloads have their last (eleventh) value as 1000 nodes. The efficiency of the proposed E²FFO algorithm mainly depends on the (Algorithm 2 point 4). The workflow energy consumption model depends on communication time, i.e., static energy, whereas computation time is related to dynamic energy. The present research proposed a static energy model to simulate an Infrastructure-as-a-Service (IaaS) environment. A virtualization platform provided by IaaS cloud to schedule scientific workflow applications. We have taken five VMs in cloud data centers and each VM has its resource capacity.

Table 2: Workflow setup

Workflow	Number of tasks										
Montage	40	70	95	125	150	400	500	750	850	900	1000
CyberShake	90	130	170	210	250	290	330	370	410	450	1000
LIGO	70	100	130	160	190	220	250	280	310	340	1000
SIPHT	100	150	250	350	450	550	650	750	850	950	1000
Epigenomics	20	60	100	140	180	220	260	300	340	380	1000

5.3 Experimental Results

This sub-section comprises two parts. The first part calculates the energy consumption and efficiency of the proposed E²FFO algorithm based on eight NAS benchmarks. The second part, E²FFO is applied to five heterogeneous scientific workflows with eleven workload classes. In the first phase, the proposed E²FFO was applied on eight parallel benchmarks (MF, CG, FT, IS, EP, BT, SP, LU) with five classes: A, B, C, S, and W. The total energy consumption for Eq. (5) was calculated based on these five classes. The experimental results are presented in Figs. 1 to 5, these results are included with the average values from several experiments for energy evaluation based on 1000 nodes. Even though we can simulate the same algorithm for fewer nodes due to lack of space, we only show evaluation results based on 1000 nodes. The results depict the different energy level consumption based on the NAS benchmarks. Figs. 1 and 2 depict that the benchmarks MG, FT, and SP, the considered classes, show no difference in energy saving up to 1000 nodes. However, CG and EP benchmarks show a small difference in energy consumption for S-W and A-B-C classes. However, there is a significant difference between S-W and A-B-C classes for IS, LU, and BT. These results may help choose the appropriate workflow for a large cloud-based application.

Figs. 3 to 5 depict that the benchmarks MG and FT, the considered classes, show no significant variations in energy-saving up to 1000 or fewer nodes. However, CG, SP, and EP benchmarks show a small difference in energy consumption for the S-W and A-B-C classes. However, there is a significant difference between S-W and A-B-C classes for IS, LU, and BT. Due to many memory requirements, experiments for bigger NAS classes than C, like D, F, and F, are not conducted.

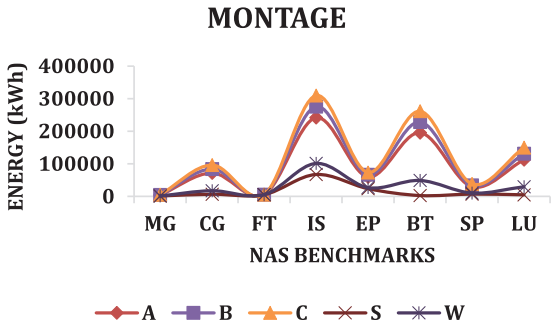

Figure 1: Running NAS benchmarks on 1000 nodes for montage workflow

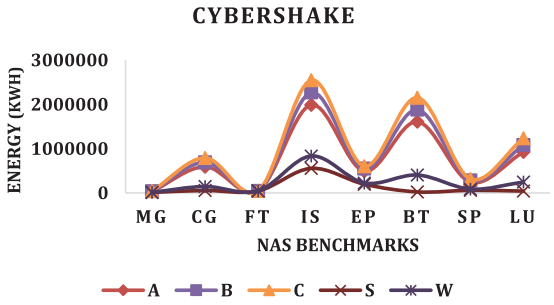

Figure 2: Running NAS benchmarks on 1000 nodes for cybershake workflow

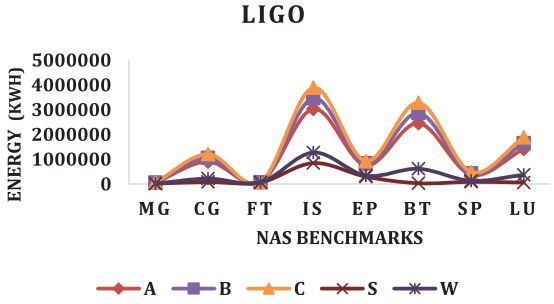

Figure 3: Running NAS benchmarks on 1000 nodes for LIGO workflow

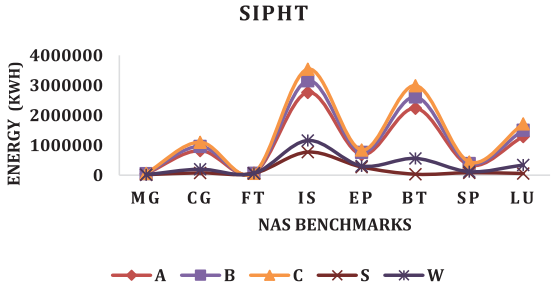

Figure 4: Running NAS benchmarks on 1000 nodes for SIPHT workflow

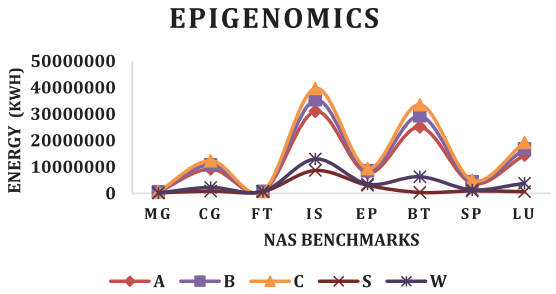

Figure 5: Running NAS benchmarks on 1000 nodes for epigenomics workflow

5.4 Comparative Results

This subsection provides the comparative analysis of the proposed E²FFO concerning EASVMC, PESVMS, EES, and HEFT algorithms. Due to resource constraints, we do not compare the traditional algorithm with E²FFO, and while tradeoff NAS benchmarks, the proposed E²FFO was compared with EASVMC, PESVMS, EES, and HEFT algorithms for five different scientific workflows. Fig. 6a depicts that up to 170 workloads, all five algorithms show almost the same behavior. In contrast, after 200 nodes, the proposed algorithms show noteworthy improvement in energy consumption for the Montage workflow. While comparing with other contemporary algorithms, the proposed E²FFO consumes an average of 15.6477%, 28.7031%, 29.9459%, and 35.9054% less energy in comparison with EASVMC, PESVMS, EES, and HEFT algorithms. Similarly, Fig. 6b for Cybeshake workflow comparison, E²FFO, EASVMC, and PESVMS show similar behavior for energy consumption, where as, in comparison with EES and HEFT algorithms, the proposed algorithm shows better results. For large applications like workloads more than 600, the E²FFO shows significant improvement in energy saving and average energy-saving 2.7992% (EASVMC), 1.9178% (PESVMS), 7.9083% (EES), and 9.6307% (HEFT).

In the case of LIGO workflow, Fig. 6c depicts that up to 130 nodes, the proposed E²FFO shows less improvement as compared to EASVMC and PESVMS, though E²FFO presents noticeable improvement concerning EES and HEFT. For large applications like workloads more than 130, E²FFO shows significant improvement in energy saving and average energy-saving 17.8928% (EASVMC), 20.7847% (PESVMS), 53.9209% (EES), and 55.6101% (HEFT). In Fig. 6d, for SIPHT workflow, the proposed algorithm (overall) shows slightly better results in comparison with EAVMC; for 200–570 nodes, the E²FFO shows significant results as compared to the PESVMS, EES, and HEFT. While taking 800–950 nodes, EES outperformed the rest four algorithms, but for 1000 or above nodes cloud application, the E²FFO saves more energy (average) as compared to EASVMC, PESVMS, EES, and HEFT with 3.1198%, 8.8413%, 6.4517%, and 7.238%, respectively. In Fig. 6e, for Epigenomics workflow, the proposed algorithm shows slightly better results in comparison with EASVMC, PESVMS, EES, and HEFT up to 160 nodes (however, 90–110 EASVMC is better than the E²FFO. After 140 or more node-based applications, the proposed E²FFO outperformed EASVMC, PESVMS, EES, and HEFT algorithms and saved significant energy with an average of 13.1302% (EASVMC), 21.267% (PESVMS), 32.5441% (EES), and 36.5302 (HEFT).

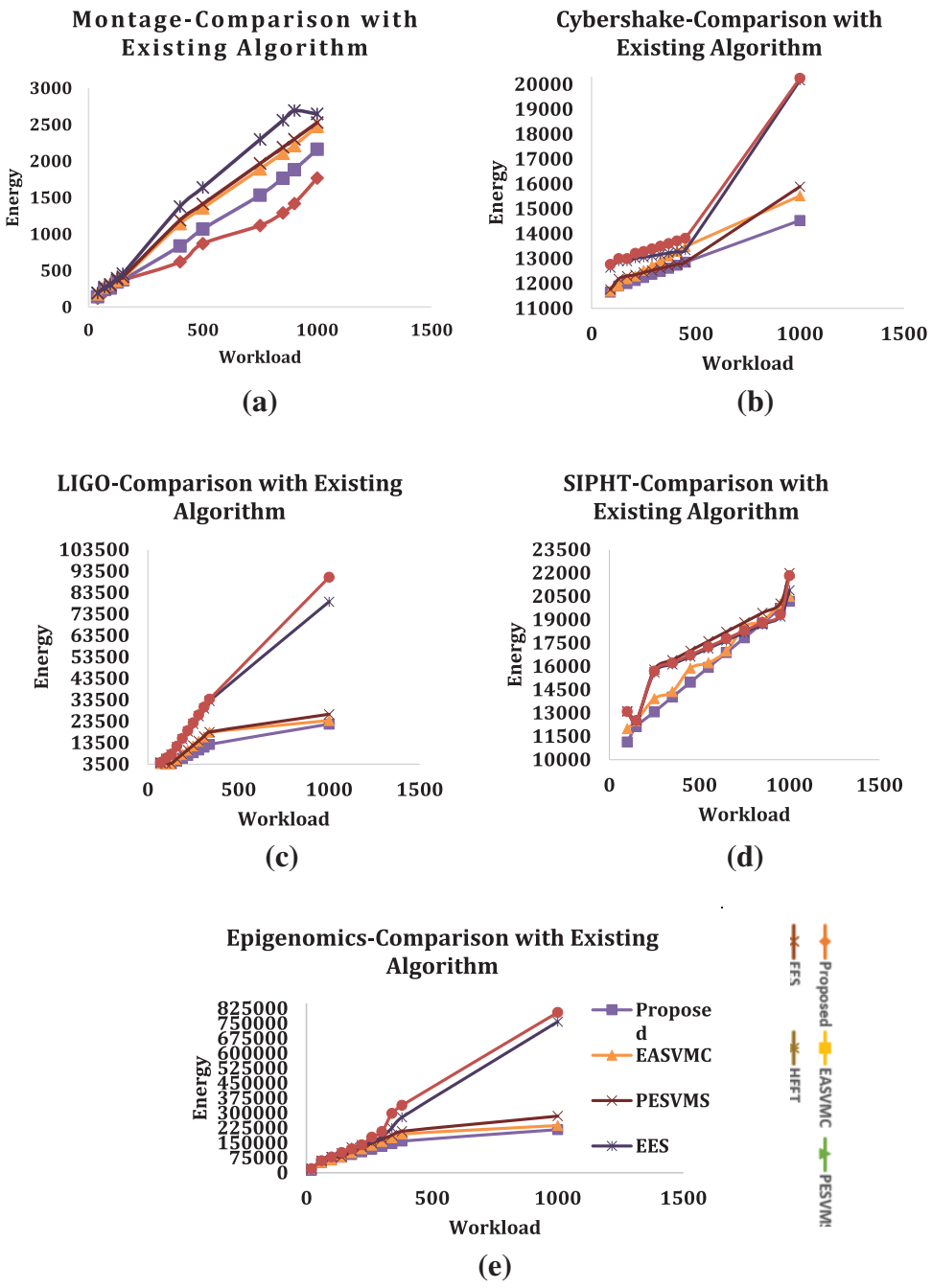

Figure 6: Proposed E²FFO performance comparison with EASVMC, PESVMS, EES, and HEFT for (a) Montage, (b) Cybershake, (c) LIGO, (d) SIPHT, (e) Epigenomics

Regarding energy consumption, the proposed E²FFO outperformed EASVMC, PESVMS, EES, and HEFT for all five mentioned workflows. Table 3 presents the percentage-wise efficiency energy consumption results, e.g., for Montage workload with workload 40–1000, the proposed E²FFO saves 15.6477% energy as compared to the EASVMC algorithm. Montage with workload capacity 40–1000,

Cybershake with 90–1000, LIGO with 70–1000, SIPHT with 100–1000, and Epigenomics with 20–1000 taken, and an overall average result for energy consumption is shown in Table 2.

Table 3: Comparative analysis of the proposed E²FFO

Proposed	EASVMC	PESVMS	EES	HEFT
Montage	15.6477	28.7031	29.9460	35.9054
CyberShake	2.7992	1.9179	7.9083	9.6307
LIGO	17.8928	20.7847	53.9210	55.6101
SIPHT	3.1199	8.8413	6.4518	7.2381
Epigenomics	13.1302	21.2670	32.5441	36.5302

The proposed E²FFO used an improved fruit fly optimization algorithm that addressed the exploration and exploitation problem while scheduling the multiple workflows. The GA approach shows an additional advantage in generating the initial workflow schedules. These schedules need to be optimized. Traditional FFO (and hybrid FFO) used the best smell values to update the location of other flies, which will be stuck into local minima. To address this problem, the proposed E²FFO uses the mean of smell concentration, i.e., the global best solution, and updates all other nodes as per the mean value. The E²FFO calculates the reduced time and static energy with the scaling factor ξ_{xy}. The energy scaling factor ξ_{xy} used to avoid exploration and exploitation during the execution.

6 Conclusions

Minimizing the carbon footprints is a worldwide concern in cloud computing. To achieve this goal efficient energy approaches are vital in optimizing energy consumption in cloud data centers. The proposed E²FFO approach uses an improved fruit-fly optimization (IFFO) algorithm to minimize energy consumption. The proposed algorithm uses selected GA for generating the initial set of solutions and the IFFO algorithm was used to reduce the overall energy of the cloud data center. Workflowsim 1.0 simulator used for simulation of the proposed algorithm. The proposed E²FFO used five scientific workflows (Montage, CyberShake, LIGO, SIPHT, and Epigenomics) to schedule three different capacity VMs (Micro, Small, and Medium). Eleven workloads from each scientific workflow were considered for better simulation.

Furthermore, the proposed E²FFO used 11 NAS benchmarks with 5 NAS classes and depicted the results. Four state-of-the-art algorithms (EASVMC, PESVMS, EES, and HEFT) were taken with five scientific workflows for comparative analysis. Table 2 showed that the proposed algorithm outperformed the existing algorithm and reduced average energy consumption from 1.91% to 55.61% compared to existing algorithms. The experimental results show that the E2FFO is 10.51% at least and 28.98% at most, better efficient for variable workloads with multiple workflows in a cloud data center. This improvement in energy optimization increases the overall system efficiency, decreases operational cost, and helps to generate less carbon footprint.

As the proposed approach worked on energy efficient parameters only, an extension of this work can be altered to sustain multiple QoS parameters such as cost, makespan, load balancing, security, latency, etc., with different magnitudes and dimensions. The initial phase of E²FFO can be enhanced to minimize the solution space in the GA. Work can be improved by handling complex scenarios of dynamic factors in a heterogeneous environment.

Acknowledgement: The authors would like to acknowledge the support of Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This reseach is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R435), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: All authors contributed equally. The authors confirm contribution to the paper as follows: Study conception and design: Ambika Aggarwal, Sunil Kumar, Ashok Bhansali; data collection: Sunil Kumar; analysis and interpretation of results: Deema Mohammed Alsekait, Diaa Salama AbdElminaam; draft manuscript preparation: Ambika Aggarwal, Sunil Kumar, Ashok Bhansali; supervision, methodology, conceptualization, formal analysis, writing-review & editing Diaa Salama AbdElminaam. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data is available with the corresponding author and can be shared on request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Aggarwal, P. Dimri, A. Agarwal, and A. Bhatt, "Self-adaptive fruit fly algorithm for multiple workflow scheduling in cloud computing environment," *Kybernetes*, vol. 24, no. 8, pp. 1704–1730, 2020. doi: 10.1108/K-11-2019-0757.

[2] J. Yu and R. Buyya, "A taxonomy of workflow management systems for grid computing," *J. Grid Comput.*, vol. 3, no. 3–4, pp. 171–200, 2005. doi: 10.1007/s10723-005-9010-8.

[3] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020. doi: 10.1126/science.aba3758.

[4] A. Javadpour, F. Ja'fari, P. Pinto, and W. Zhang, "Mapping and embedding infrastructure resource management in software-defined networks," *Clust. Comput.*, vol. 26, no. 1, pp. 461–475, 2023. doi: 10.1007/s10586-022-03789-8.

[5] A. Katal, S. Dahiya, and T. Choudhury, "Energy efficiency in cloud computing data centers: A survey on software technologies," *Cluster Comput.*, vol. 26, no. 3, pp. 1845–1875, 2023. doi: 10.1007/s10586-022-03713-0.

[6] P. Chithaluru, S. Kumar, A. Singh, A. Benslimane, and S. K. Jangir, "An energy-efficient routing scheduling based on fuzzy ranking scheme for Internet of Things (IoT)," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7251–7260, 2021. doi: 10.1109/JIOT.2021.3098430.

[7] H. Mikram, S. E. Kafhali, and Y. Saadi, "HEPGA: A new effective hybrid algorithm for scientific workflow scheduling in cloud computing environment," *Simul. Model. Pract. Theory.*, vol. 130, pp. 102864, 2024. doi: 10.1016/j.simpat.2023.102864.

[8] J. D. Ullman, "NP-complete scheduling problems," *J. Comput. Syst. Sci.*, vol. 10, no. 3, pp. 384–393, 1975. doi: 10.1016/S0022-0000(75)80008-0.

[9] A. Javadpour, A. Nafei, F. Ja'fari, P. Pinto, W. Zhang and A. K. Sangaiah, "An intelligent energy-efficient approach for managing IoE tasks in cloud platforms," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 4, pp. 3963–3979, 2022. doi: 10.1007/s12652-022-04464-x.

[10] M. Mokni, S. Yassa, J. E. Hajlaoui, M. N. Omri, and R. Chelouah, "Multi-objective fuzzy approach to scheduling and offloading workflow tasks in Fog-Cloud computing," *Simul. Model. Pract. Theory*, vol. 123, pp. 102687, 2023. doi: 10.1016/j.simpat.2022.102687.

[11] M. Mezmaz, N. Melab, and E. G. Talbi, "An efficient load balancing strategy for grid-based branch and bound algorithm," *Parallel Comput.*, vol. 33, no. 4–5, pp. 302–313, 2007. doi: 10.1016/j.parco.2007.02.004.
[12] H. Topcuoglu, S. Hariri, and M. Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 3, pp. 260–274, Mar. 2002. doi: 10.1109/71.993206.
[13] N. A. Bahnasawy, M. A. Koutb, M. Mosa, and F. Omara, "A new algorithm for static task scheduling for heterogeneous distributed computing systems," *Afr. J. Math. Comput. Sci. Res.*, vol. 3, no. 6, pp. 221–234, 2011. doi: 10.11591/eei.v5i3.559.
[14] N. A. Bahnasawy, F. Omara, M. A. Koutb, and M. Mosa, "Optimization procedure for algorithms of task scheduling in high performance heterogeneous distributed computing systems," *Egypt. Inform. J.*, vol. 12, no. 3, pp. 219–229, 2011. doi: 10.1016/j.eij.2011.10.001.
[15] M. Adhikari and T. Amgoth, "Efficient algorithm for workflow scheduling in the cloud computing environment," presented at the Ninth Int. Conf. Contemp. Comput. (IC3), Noida, UP, India, Aug. 2016. doi: 10.1177/1550147720949142.
[16] M. Sardaraz and M. Tahir, "A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing," *Int. J. Distrib. Sens. Netw.*, vol. 16, no. 8, 2020. doi: 10.1177/1550147720949142.
[17] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, Massachusetts, USA: MIT Press, 1992.
[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at the Int. Conf. Neural Netw., Perth, WA, Australia, Perth, WA, Australia, Nov. 27–Dec. 1, 1995.
[19] A. M. Manasrah and H. B. Ali, "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," *Wirel. Commun. Mob. Comput.*, vol. 2018, pp. 1–16, 2018. doi: 10.1155/2018/1934784.
[20] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, "A GSA-based hybrid algorithm for bi-objective workflow scheduling in cloud computing," *Future Gener. Comput. Syst.*, vol. 83, pp. 14–26, 2018. doi: 10.1016/j.future.2018.01.005.
[21] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei and S. Hu, "Minimizing cost and makespan for workflow scheduling in the cloud using fuzzy dominance sort based HEFT," *Future Gener. Comput. Syst.*, vol. 93, pp. 278–289, 2019. doi: 10.1016/j.future.2018.10.046.
[22] A. Bhardwaj and C. R. Krishna, "Impact of factors affecting pre-copy virtual machine migration technique for cloud computing," *Mater. Today Proc.*, vol. 18, pp. 1138–1145, 2019. doi: 10.1016/j.matpr.2019.06.574.
[23] J. J. Durillo, V. Nae, and R. Prodan, "Multi-objective energy-efficient workflow scheduling using list-based heuristics," *Future Gener. Comput. Syst.*, vol. 36, pp. 221–236, 2014. doi: 10.1016/j.future.2013.07.005.
[24] F. Cao, M. M. Zhu, and C. Q. Wu, "Energy-efficient resource management for scientific workflows in clouds," presented at the IEEE World Congr. Serv., Anchorage, AK, USA, Jun. 23–27, 2014. doi: 10.1109/SERVICES.2014.76.
[25] K. Bousselmi, Z. Brahmi, and M. M. Gammoudi, "Energy efficient partitioning and scheduling approach for scientific workflows in the cloud," presented at the IEEE Int. Conf. Serv. Comput. (SCC), San Francisco, CA, USA, Jun. 27–Jul. 2, 2016.
[26] H. Chen, X. Zhu, D. Qiu, H. Guo, L. T. Yang and P. Lu, "EONS: Minimizing energy consumption for executing real-time workflows in virtualized cloud data centers," presented at the 45th Int. Conf. Parallel Process. Workshops (ICPPW), Philadelphia, PA, USA, Aug. 16–19, 2016.
[27] A. Rehman, S. S. Hussain, Z. U. Rehman, S. Zia, and S. Shamshirband, "Multi-objective approach of energy efficient workflow scheduling in cloud environments," *Concurr. Comput.: Pract. Exper.*, vol. 31, no. 8, pp. e4949, 2019. doi: 10.1002/cpe.4949.
[28] R. Garg and M. Mittal, "Reliability and energy efficient workflow scheduling in cloud environment," *Clust. Comput.*, vol. 22, no. 4, pp. 1283–1297, 2019. doi: 10.1007/s10586-019-02911-7.
[29] W. Ahmad, B. Alam, and A. Atman, "An energy-efficient big data workflow scheduling algorithm under budget constraints for heterogeneous cloud environment," *J. Supercomput.*, vol. 77, pp. 11946–11985, 2021. doi: 10.1007/s11227-021-03733-4.

[30] M. Hussain, L. F. Wei, A. Lakhan, S. Wali, S. Ali and A. Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing," *Sustain. Comput.: Inform. Syst.*, vol. 30, pp. 100517, 2021. doi: 10.1016/j.suscom.2021.100517.

[31] R. Medara and R. S. Singh, "Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization," *Simul. Model. Pract. Theory*, vol. 110, pp. 102323, 2021. doi: 10.1016/j.simpat.2021.102323.

[32] N. Mohanapriya, G. Kousalya, P. Balakrishnan, and C. P. Raj, "Energy efficient workflow scheduling with virtual machine consolidation for green cloud computing," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1561–1572, 2018. doi: 10.1016/j.simpat.2021.102323.

[33] Q. Huang, S. Su, J. Li, P. Xu, K. Shuang and X. Huang, "Enhanced energy-efficient scheduling for parallel applications in cloud," presented at the 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput., Ottawa, ON, Canada, May 13–16, 2012.

[34] P. Verma and Afsaruddin, "Review on workflow scheduling in cloud environment: A comprehensive study," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 9, no. 2, pp. 84–88, 2021. doi: 10.21276/ijircst.2021.9.2.12.

[35] X. Wu and A. Che, "Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search," *Omega*, vol. 94, p. 102117, 2020. doi: 10.1016/j.omega.2019.102117.

[36] D. Soni, D. Srivastava, A. Bhatt, A. Aggarwal, S. Kumar and M. A. Shah, "An empirical client cloud environment to secure data communication with alert protocol," *Math. Probl. Eng.*, vol. 2022, pp. 1–14, 2022. doi: 10.1155/2022/4696649.

[37] A. Aggarwal, S. Kumar, A. Bhatt, and M. A. Shah, "Solving user priority in cloud computing using enhanced optimization algorithm in workflow scheduling," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, 2022. doi: 10.1155/2022/7855532.

[38] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurr. Comput.: Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, 2012. doi: 10.1002/cpe.1867.

[39] X. F. Liu, Z. H. Zhan, J. D. Deng, Y. Li, T. Gu and J. Zhang, "An energy-efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, 2016. doi: 10.1109/TEVC.2016.2623803.

[40] A. Aggarwal, P. Dimri, and A. Agarwal, "Survey on scheduling algorithms for multiple workflows in cloud computing environment," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 6, pp. 565–570, 2019. doi: 10.26438/ijcse/v7i6.565570.

[41] M. V. S. Masdari, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 60, pp. 64–82, 2016. doi: 10.1016/j.jnca.2016.01.018.

[42] M. Adhikari, T. Amgoth, and S. N. Srirama, "A survey on scheduling strategies for workflows in cloud environment and emerging trends," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, 2018. doi: 10.1145/3325097.