International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Effective Workflow Scheduling in Cloud Platform Using Data Aware based Adaptive Gravitational Search Algorithm

Vinay Kumar Sriperambuduri[1]* **Nagaratna Medishetty[1]**

[1]*Department of Computer Science and Engineering, JNTUH College of Engineering, Hyderabad, India*
* Corresponding author's Email: s.vinaykumar@staff.vce.ac.in

Abstract: In recent periods, cloud-computing environments are widely utilizing scientific workflows for executing large-scale applications. The workflow scheduling with the scientific standard for optimizing quality of service (QoS) parameters is a hard task. In the existing research studies, several metaheuristics optimization algorithms are employed for satisfying the QoS parameters such as resource utilization, cost, and makespan. Still, the existing metaheuristics optimization algorithms are insignificant for maintaining the balance between exploitation and exploration in a search space, because the algorithms are easily trapped in local optima. For addressing the above-stated issues, a data aware based adaptive gravitational search algorithm (DA-AGSA) technique is implemented to minimize the cost and makespan, and to schedule workflows in the cloud-computing platform. In the conventional GSA technique, a random coefficient is replaced by an adaptive weight function for improving convergence rate, and further, the weight function is multiplied with an acceleration term for facilitating quicker convergence. In this article, the performance of the DA-AGSA technique is validated by utilizing workflow sim for scheduling multiple workflows. An extensive experimental investigation showed that the DA-AGSA technique almost reduced 20% of the cost and 15% of the makespan compared to the conventional optimization algorithms on the Montage, CyberShake, and Epigenomics workflows with 1000 tasks. In addition, the DA-AGSA technique achieved a reliability of 0.99, 0.98, and 0.98 on the Montage, CyberShake and Epigenomics workflows with 1000 tasks.

Keywords: Cloud computing, Gravitational search algorithm, Data aware scheduling, Machine learning, Workflow scheduling.

1. Introduction

In recent decades, cloud computing has delivered varied services to users through the internet, which is used for implementing dissimilar commercial applications [1]. However, computing includes numerous techniques such as grid, parallel, and distributed computing [2-3]. Generally, cloud computing involves multiple technologies that create a new way of handling information technology (IT). Cloud computing offers highly available and elastically scalable resources as subscription-based services like utility computing to execute scientific workflows [4-5]. The primary aim of the task scheduling methods is to increase the acceleration of the execution, where it allocates the resources to the workloads that have different execution times [6]. The proper allocation of resources effectively balances the workload and is further classified into dynamic and static methodologies [7]. Cloud workload scheduling provides an effective mapping between the resources and tasks, whereas a significant scheduling algorithm maintains an effective trade-off between resource utilization and user requirements [8].

Usually, cloud task scheduling is a nondeterministic polynomial (NP) hard optimization problem; therefore, several optimization algorithms are developed for addressing the aforementioned problem [9-10]. In most cases, the traditional metaheuristics-based optimization algorithms result in a higher computational time [11]. In addition, the optimal solution is obtained only by exploring a larger search region; in this case, the workflow should be well managed and defined [12-13]. In order to overcome the aforementioned problems, a novel DA-AGSA technique is proposed in this manuscript for effective workflow scheduling in cloud environments. The proposed DA-AGSA technique improves data aware scheduling by determining the best resource or node for scheduling a task and best order of tasks to be executed. In this study, the effectiveness of the proposed DA-AGSA technique is analysed using reliability, cost and makespan on Montage, CyberShake and Epigenomics workflows with 1000 tasks. The experimental results represent that the proposed DA-AGSA technique has significantly reduced cost and makespan in workflow scheduling when compared to the existing optimization algorithms with better reliability.

The research papers related to "workflow scheduling in the cloud computing platform" are briefly surveyed in section 2. The methodology explanation and the simulation outcomes of the DA-AGSA technique are given in sections 3, and 4. The conclusion of the study is mentioned in section 5.

2. Related works

Qin [14] introduced a hybrid collaborative multi-objective fruit-fly optimization algorithm (HCMFOA) for optimizing the cost and running time in cloud environments. The introduced HCMFOA utilized a reference point-based clustering technique for dividing a single swarm into multi-sub-swarms. In this study, the hybridization includes two rules namely assignments and non-linear weight vectors, which were utilized for initializing the fruit-flies-location in the search space. Additionally, three neighborhood operations were carried out in the collaborative-smell-based-foraging and the crossover operator was utilized for performing exploitations in the local regions. The extensive experimental outcomes demonstrated that the introduced HCMFOA achieved better performance related to the prior state-of-the-art models. In addition to this, Aggarwal [15] introduced an improved FOA for minimizing the cost and makespan in the cloud platform to schedule multiple workflows. As depicted in the resulting section, the developed improved FOA outperformed the existing optimization algorithms in terms of cost and makespan. However, the metaheuristic-based optimization algorithms like HCMFOA and improved FOA consist of concerns like higher dimensional non-linear optimization and being trapped into a local optimum at the later evolutional phases.

Loheswaran [16] implemented an upgraded FOA to optimize the resource management and task scheduling processes. The implemented upgraded FOA was related to the existing optimization algorithms and the obtained experimental outcomes show that the implemented algorithm was better than the existing optimization algorithms in terms of resource utilization and task allocation. Additionally, Fu [17] developed a multi-objective discrete FOA for workflow scheduling in the cloud environment. In this study, the multi-objective discrete FOA consists of 5-search approaches genetic search, vision search, smell search, solution representation and heuristic decoding rules. In the resulting section, the experimental investigation was conducted on 25 instances. The obtained experimental outcome showed that the presented multi-objective discrete FOA performs more effectively on the 25 instances than its peers. The scheduling process was complex in the conventional FOA, so, a novel hybrid machine-learning model can be included with the presented system to further enhance scheduling mechanism.

Aziza and Krichen [18] used genetic algorithms based on heterogeneous earliest finish times for workflow scheduling in the cloud platform. The experiments conducted on the real-time workflow databases demonstrated that the developed algorithm achieved maximum performance than the other optimization strategies. The presented strategy has not focused on the power consumption of data centers, which needs to be concentrated while planning workflows in the cloud environments. Abualigah and Diabat [19] implemented a hybrid Ant-Lion optimization (ALO) algorithm to solve task scheduling issues in the cloud platforms. In this study, the hybrid ALO superiorly increases resource utilization and reduces the makespan. Here, the ALO comprises an elite-based differential evolution methodology for improving exploitation and exploration ability. Additionally, Saeedi [20] implemented an improved multi-objective Particle swarm optimization (PSO) algorithm in a cloud environment for minimizing energy consumption, cost and makespan, and maximizing reliability. In this application, the conventional ALO and PSO algorithms were trapped in local optima.

Konjaang [21] has presented a multi-objective workflow optimization strategy (MOWOS) in cloud computing platform. The efficacy of the developed MOWOS was validated by means of execution cost. In addition to this, Zeedan [22] presented a hybrid optimization algorithm named enhanced binary artificial bee colony with pareto front (EBABC-PF) for effective workflow scheduling in the cloud environment. However, the higher execution cost and processing cost were the major concerns in the

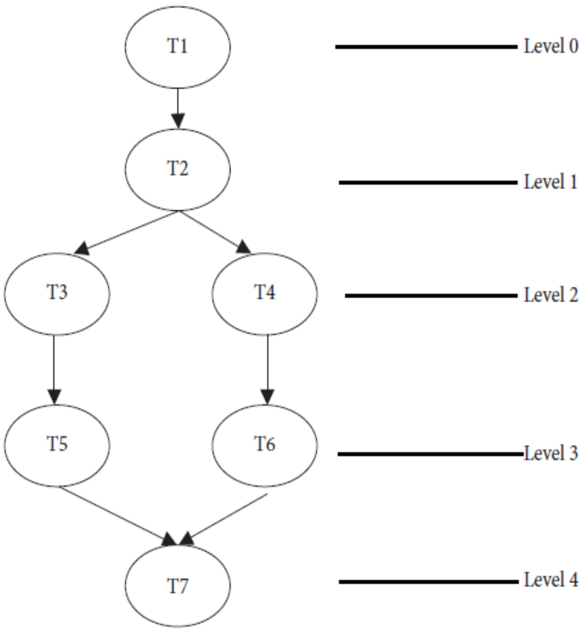

Figure. 1 Graphical presentation of sample workflow

existing MOWOS and EBABC-PF models. To address the aforementioned issues, a novel DA-AGSA technique is implemented for workflow scheduling in cloud computing platforms to reduce the cost and makespan parameters.

3. Methodology

In this research manuscript, the performance of the proposed DA-AGSA technique is tested on the Montage, CyberShake and Epigenomics workflows. The CyberShake workflow is utilized in earthquake science for epitomizing earthquake hazards by creating synthetic seismograms. Further, the Montage workflow is one of the astronomical applications, which is generated by the National Aeronautics and Space Administration/Infrared Processing and Analysis Centre. In addition, the Epigenomics workflows are created by the Pegasus team and USC Epigenomics center to automate several operations in genome sequence processing.

3.1 Problem definition

In cloud computing environments, large scale applications are deployed in the form of workflows $W = (T, E)$. Pictorially, the workflows are stated utilizing a Directed Acyclic-Graph, and the term $T = \{T_1, T_2, \dots T_n\}$ is represented as tasks. Generally, the applications are categorized into numerous independent and dependent subtasks. Fig. 1 represents a sample workflow. With reference to Fig. 1, the whole application is partitioned into 7 sub-tasks like $T_1, T_2, \dots . T_7$, and it falls between 5 dissimilar levels (levels 0 to 4). In level 2, the tasks T_3 and T_4 are independent because these two tasks are at similar levels. So, the tasks T_3 and T_4 are executed subsequently on dissimilar resources [23-24].

Once the execution of a task T_2 is completed at level 1, the tasks T_3 and T_4 are executed with the output data of the task T_2. Similarly, the task T_2 execution depends on the task T_1. In this scenario, the IaaS cloud provider is considered for resource heterogeneity, where various Virtual Machines (VMs) are available with dissimilar configurations. The 2-dimensional bid is achieved by combining the cloud marketplace with the cloud service provider, and it is defined in Eq. (1).

$$B_{VMi} = (P_{VMi}, C_{VMi}) \tag{1}$$

Where, C_{VMi} represents execution cost and P_{VMi} states the processing capacity of the VMs. On a resource VM_l , the execution time of a task is computed by utilizing Eq. (2).

$$ET_j^l = \frac{S_{T_j}}{\left(P_{VM_l} \times \left(1 - P_{var_l}\right)\right)} \tag{2}$$

Where, ET_j^l represents execution time, $T_j = \{T_1, T_2, T_3, \dots T_J\} \forall \{1,2,3, .. J\}$ indicates a task, S_{T_j} represents the size of a task which is in bytes and, and P_{var_l} states processing capacity of VM represented in Million Instructions per Second (MIPS). Usually, the workflow involves task dependency, the transfer time of data DT_{jk} is computed using Eq. (3).

$$DT_{jk} = \frac{D_{out_{T_j}}}{bw} \tag{3}$$

Where, D_{out} represents generated data and the bandwidth between every VM is represented as bw. Further, the data transferring rate between the scheduled tasks on a similar resource is zero. On a resource VM_l, the total processing time PT_j^l of every task T_j is computed using Eq. (4).

$$PT_j^l = ET_j^l + (\sum_l^e DT_{jk} \times Q) \tag{4}$$

Where, e represents the number of edges inter-connected with every task T_j, and $Q = 0$, if two tasks are scheduled on similar VMs, else one [25-27]. In this manuscript, the DA-AGSA technique is used to find the optimal schedule in the workflows: Montage, CyberShake and Epigenomics, for minimizing the total execution time and cost.

3.2 Workflow scheduling using DA-AGSA technique

Most of the conventional metaheuristics-based optimization algorithms suffer from high-dimensional non-linear optimization issues. The AGSA technique divides the population in an effective manner, where each population group determines dissimilar possible solutions in a run of the heuristic search.

The optimization algorithm with data aware scheduling allows the user to minimize the end-to-end workflow turn-around time successfully [28]. First, data aware scheduling is one of the effective techniques utilized in distributed computing systems for scheduling tasks based on the data location and data dependencies. The primary objective of the data aware scheduling is to reduce the communication costs and data movements between the nodes in a system by scheduling the tasks on the similar rack or node. The data aware scheduling significantly enhances the efficiency and performance of the system by decreasing the time needed to access data and network traffic. The optimization algorithm in data aware scheduling effectively makes decisions about where to schedule the tasks based on the data location and data dependencies, and also makes decisions about the order in which tasks should be executed.

In the developed DA-AGSA technique, based on Newton's gravity law, the particles are attracted to other particles with the usage of "Gravitational Force", which is directly proportional to the mass produced and inversely proportional to the square of distance between the particles [29]. The "Gravitational Force" F is mathematically represented in Eq. (5) [30]. In the developed DA-AGSA technique, the gravitational constant value G is decreased with age and it is computed utilizing Eq. (6) and Eq. (7).

$$F = G\frac{M_1 M_2}{R^2} \tag{5}$$

Where, $M_1 and\ M_2$ are represented as the mass of the 1st and 2nd particles, and R^2 is stated as the distance between the two particles.

$$G(t) = G(t_0) \times (\frac{t_0}{t})^{\beta}, \beta < 1 \tag{6}$$

$$G(t+1) = G(t) exp\ (-\frac{\alpha t}{T}) \tag{7}$$

In the DA-AGSA technique, each agent or mass is determined by its passive gravitational mass, inertial mass, active gravitational mass, and position. By considering the aspects of the aforementioned masses, newton's law is updated, as mentioned in Eq. (8).

$$F_{ij} = G\frac{M_{aj} \times M_{pi}}{R^2} \tag{8}$$

Where, $a_i = \frac{F_{ij}}{M_{ii}}$

By considering a network with N masses (agents), the position of the i^{th} agent is defined, and it is mathematically denoted in Eq. (9). In a particular time t, the force that acts on a mass i from a mass j is determined using Eq. (10) and the Euclidean distance between two search agents i and j is computed by utilizing Eq. (11).

$$X_i = (x_i^1, \ldots., x_i^d, \ldots.. x_i^n), for\ i = 1,2,3, \ldots., N \tag{9}$$

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon}[x_j^d(t) - x_i^d(t)] \tag{10}$$

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \tag{11}$$

For accounting stochastic characteristics to the optimization technique, the force that acts on the agent i is assumed as the weighted sum of d_{th} force components in a dimension d, and it is exerted from another agent, which is mathematically specified in Eq. (12).

$$F_i^d(t) = \sum_{j=1, j\neq 1}^{N} rand_j F_{ij}^d(t) \tag{12}$$

The conventional GSA technique is trapped into local optima, where this problem is avoided by performing the exploration search at the beginning. The conventional GSA technique's efficiency is further improved, if the $Kbest$ agents are attracted with other search agents, and it is mathematically expressed in Eq. (13).

$$F_i^d(t) = \sum_{j \epsilon Kbest, j\neq 1} rand_j\ F_{ij}^d(t) \tag{13}$$

Where, $rand_j$ indicates random coefficient value. The velocity $v_i^d(t)$ and the position $x_i^d(t+1)$ of the agents are computed utilizing Eq. (14) and Eq. (15).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{14}$$

$$v_i^d(t) = rand_i \times v_i^d(t) + a_i^d(t) \tag{15}$$

At the beginning, the gravitational constant value G is initialized that increases the search accuracy. The G is expressed as a function with time t, and initial value G_0, as mentioned in Eq. (16).

$$G(t) = G(G_0, t) \tag{16}$$

The gravitational $m_i(t)$ and the inertial mass $M_i(t)$ is updated, as stated in Eq. (17), Eq. (18), and Eq. (19).

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1,2,3, \dots N \tag{17}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{18}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{19}$$

Where, $fit_i(t)$ specifies fitness value of an agent i at a time interval t. The objective $fit_i(t)$ function values are specified in Eq. (20) and Eq. (21). The minimization issue $worst(t)$ and $best(t)$ are mathematically denoted in Eq. (22) and Eq. (23).

$$Makespan = max\ (machine\ bound[j]) \tag{20}$$

$$Cost = sum\ of\ fixed\ cost[j]\ if\ operation\ [j] > 0 \tag{21}$$

$$worst(t) = max_{j\epsilon\{1,2....N\}} fit_j(t) \tag{22}$$

$$best(t) = min_{j\epsilon\{1,2....N\}} fit_j(t) \tag{23}$$

On the other hand, the random coefficient value $rand_j$ in Eq. (15) is replaced by the adaptive weight functions w_1 and w_2 in order to improve the convergence rate of the conventional GSA technique. The mathematical expressions of the weight functions w_1 and w_2 are stated in Eq. (24) and (25).

$$w_1 = w_i^t = \left| \frac{min\ (fit_i^t, mean(fit^t))}{max\ (fit_i^t, mean(fit^t))} \right| \tag{24}$$

$$w_2 = w^t = w_{min} - t \times \frac{w_{max} - w_{min}}{T} \tag{25}$$

Where, w_{min} and w_{max} are indicated as user defined parameters. The updated velocity $v_i^d(t)$ is given in Eq. (26).

$$v_i^d(t) = w_2 \times v_i^d(t) + w_1 \times a_i^d(t) \tag{26}$$

The adaptive weight functions guide convergence of the optimization algorithm as the solutions to exhibit precise movements, when moving towards the global optimum. The parameter-settings of the DA-AGSA technique are: total number of iterations is 100, $\alpha = 20$, $\beta = 10$, and G_0 is 100. In addition, the steps involved in the DA-AGSA technique are given below:

Steps involved in the DA-AGSA technique

1. Performed AGSA in data aware workflow scheduling
2. In AGSA, randomly initialize the population and replace random coefficient by an adaptive weight function
3. Compute worst and best fitness values
4. For every agent, do:
    Compute fitness
    Compute mass
    Compute mass force
    Compute mass acceleration
    Update mass velocity
    Identify new position of an agent
   End For
5. If the stopping criteria is not met, then, again go to Step 2, else stop.

4. Simulation results

In this research manuscript, the experimental investigation of the DA-AGSA technique is performed utilizing the workflow sim framework on a computer with 16GB random access memory, 4TB hard disk, 3.2 GHz computer processing unit and the Win-10 (64-bit) operating system. In this manuscript, the performance of the developed DA-AGSA technique is validated on the Montage, CyberShake and Epigenomics workflows with 1000 tasks and the experimental results are compared with five metaheuristics optimization algorithms like FOA, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithm (GA) and DA-GSA. The effectiveness of the DA-AGSA technique is investigated by means of makespan, cost, and reliability. The makespan of a workflow is determined as the latest finished time on all the VMs and the cost is determined by multiplying the task duration of a task to the allocated VMs price for all the tasks. The reliability is defined as the probability of task execution over the allocated processor successfully without errors.

4.1 Quantitative analysis

In this sub-section, the developed DA-AGSA technique is compared with FOA, ACO, PSO, GA and DA-GSA based on two scheduling objectives

Received: April 19, 2023. Revised: May 21, 2023.

Table 1. Experimental results of the developed DA-AGSA technique in terms of cost and makespan

Workflows	Measures	Optimization techniques					
		FOA	PSO	GA	ACO	DA-GSA	DA-AGSA
Montage	Cost	5142.99	5034.99	5584.80	4604.80	4590.45	4322.43
	Makespan	4827.77	5115.77	4900.46	4778.23	4678.44	4549.70
CyberShake	Cost	52231.33	50983.04	56312.87	46288.62	43234.90	41384.92
	Makespan	158819.33	155981.76	159688.11	144670.71	139393.43	126495.48
Epigenomics	Cost	553203.16	529891.16	583102.87	470317.95	460943.83	451568.86
	Makespan	208308.05	185860.05	194636.14	168008.04	154345.81	144146.80

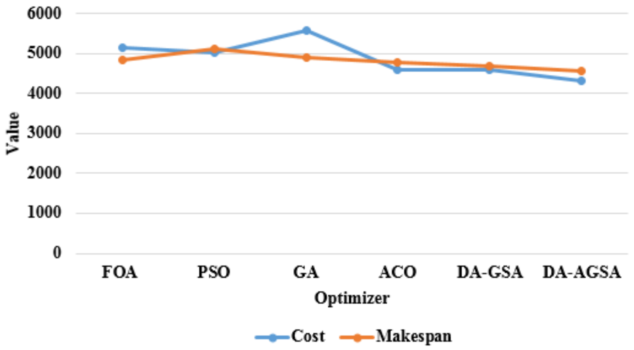

Figure. 2 Makespan and cost analysis on Montage workflow

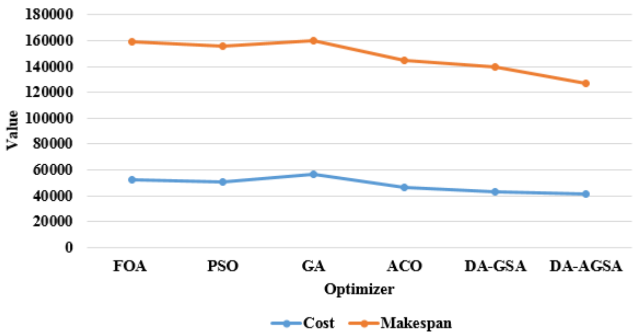

Figure. 3 Makespan and cost analysis on CyberShake workflow

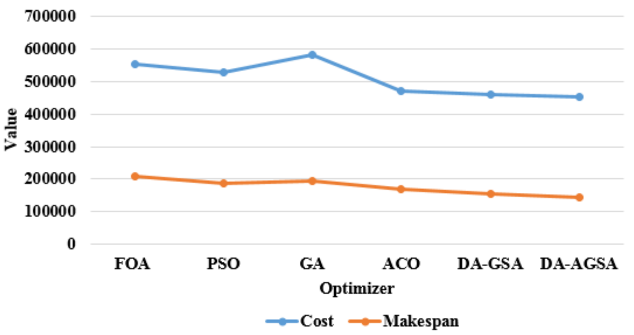

Figure. 4 Makespan and cost analysis on Epigenomics workflow

such as cost and makespan. In this study, the optimization algorithms are executed for 100 iterations with 1000 tasks, and the experimental results state that the DA-AGSA obtained high performance when compared to the existing algorithms in light of cost and makespan. Experimental results are represented in the Fig. 2, Fig. 3, and Fig. 4 for Montage, CyberShake and Epigenomics workflows. Table 1 clearly denotes that the DA-AGSA technique outperformed the existing algorithms like FOA, ACO, PSO, GA, and DA-GSA in both parameters.

In addition, the percentage-wise improvement of the developed DA-AGSA technique is represented in Fig. 5 and Table 2. On the Montage workflow, the DA-AGSA technique is 18.98%, 16.48%, 29.20%, 6.53% and 6.20% better compared to the existing optimization techniques like FOA, PSO, GA, ACO, and DA-GSA in light of cost. Correspondingly, the DA-AGSA technique showed 6.11%, 12.44%, 7.70%, 5.02%, and 2.82% improvement in terms of makespan related to other optimization algorithms.

On the CyberShake workflow, the improvement percentage is 26.20%, 23.19%, 36.07%, 11.84%, and 4.47% in light of cost, and 25.55%, 23.31%, 26.24%, 14.36%, and 10.19% by means of makespan related to other algorithms like FOA, PSO, GA, ACO, and DA-GSA.

Correspondingly, in the Epigenomics workflow, the implemented DA-AGSA technique showed an improvement of 22.50%, 17.34%, 29.12%, 4.15%, and 2.07% in light of cost, and 44.51%, 28.93%, 35.02%, 16.55%, and 7.07% in terms of makespan related to the FOA, PSO, GA, ACO, and DA-GSA algorithms. The experimental examination showed that the developed DA-AGSA technique is more effective in optimizing the parameters related to other algorithms.

In addition to this, the implemented DA-AGSA technique has achieved reliability of 0.99, 0.98 and 0.98 on the Montage, CyberShake and Epigenomics workflows with 1000 tasks. The attained reliability results are better when compared to the optimization algorithms.

4.2 Comparative analysis

In this section, initially comparative evaluation is carried out between the DA-AGSA technique and the existing MOWOS technique [21], which is implemented by J.K. Konjaang, and L. Xu. As

Table 2. Percentage-wise results of the developed DA-AGSA technique over the existing algorithms in terms of cost and makespan

Workflows	Measures	Optimization techniques				
		FOA (%)	PSO (%)	GA (%)	ACO (%)	DA-GSA (%)
Montage	Cost	18.98	16.48	29.20	6.53	6.20
	Makespan	6.11	12.44	7.70	5.02	2.82
CyberShake	Cost	26.20	23.19	36.07	11.84	4.47
	Makespan	25.55	23.31	26.24	14.36	10.19
Epigenomics	Cost	22.50	17.34	29.12	4.15	2.07
	Makespan	44.51	28.93	35.02	16.55	7.07

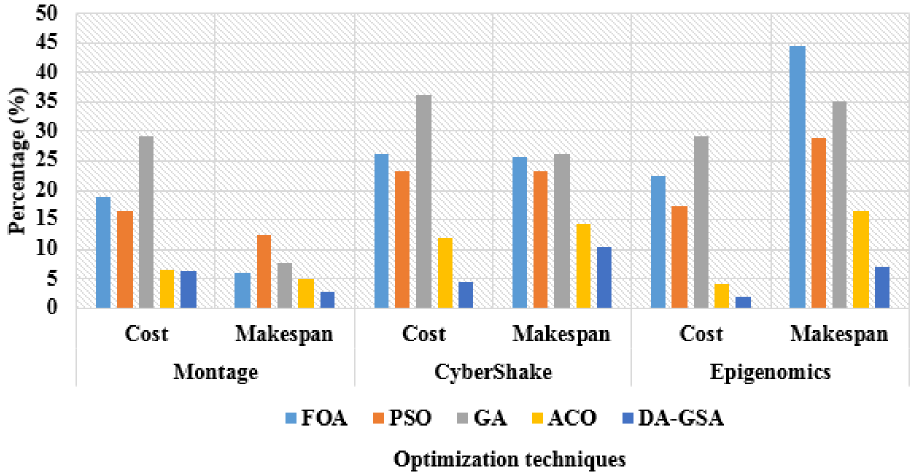

Figure. 5 Percentage-wise comparison of the DA-AGSA technique over the existing algorithms in terms of cost and makespan

Table 3. Comparative evaluation between the DA-AGSA technique and MOWOS

Models	Workflows	Workflow tasks	Execution cost
MOWOS [21]	Montage	1000	5,000
	CyberShake		62,000
DA-AGSA	Montage		4,590.45
	CyberShake		43,234.90

depicted in the resulting section, the MOWOS has an execution cost of 5,000 and 62,000 on the Montage and CyberShake workflows with 1000 tasks. Related to this existing model, the proposed DA-AGSA technique has a lower cost of 4,590.45 and 43,234.90 on the Montage and CyberShake workflows with 1000 tasks, and it is indicated in Table 3.

On the other hand, Zeedan [22] developed a hybrid optimization algorithm named EBABC-PF for effective workflow scheduling in cloud environments. As stated in Table 4, the performance analysis was carried out with different tasks (100 and 1000) and VMs (20 and 80) on the Montage, CyberShake, and Epigenomics workflows by means of processing cost. Here, the cost of each VM is 2.224 ($\$/Hr$). As seen in Tables 4 and 3, the proposed DA-AGSA technique has lower processing cost and execution cost than the comparative techniques. The extensive experimental investigation states that the DA-AGSA technique has significantly solved the issue of high cost, which was a major concern stated in the literature section.

5. Conclusion

In recent decades, workflow scheduling has played a crucial part in cloud-computing applications and environments. Several nature-inspired optimization algorithms are employed for workflow scheduling in the cloud platform. The existing optimization algorithms fall highly into local optima for maintaining the balance between exploitation and exploration spaces. In this manuscript, a new DA-AGSA technique is implemented for workflow scheduling that effectively reduces the makespan and cost parameters with optimal solutions. Here, the proposed DA-AGSA technique is implemented on a workflow sim platform, and the simulation result is analysed on the Montage, CyberShake and Epigenomics workflows. The experimental result on Montage workflow indicates that the developed DA-AGSA technique is superior compared to the FOA, PSO, GA, ACO, and DA-GSA algorithms by 18.98%, 16.48%, 29.20%, 6.53% and 6.20% in terms of cost, and 6.11%, 12.44%, 7.70%, 5.02%, and 2.82% by

Table 4. Comparative evaluation between the DA-AGSA technique and EBABC-PF technique

Processing cost							
Tasks	**VMs**	**Montage**		**CyberShake**		**Epigenomics**	
		EBABC-PF [22]	DA-AGSA	EBABC-PF [22]	DA-AGSA	EBABC-PF [22]	DA-AGSA
100	20	653.11	432.36	959.89	654.31	1459.89	834.31
100	80	1525.87	624.22	1994.69	822.11	2629.70	1022.07
1000	20	1649.30	532.75	2205.16	755.99	2539.53	966.31
1000	80	3294.64	742.06	3447.45	934.87	3302.38	1140.07

means of makespan. The DA-AGSA technique showed better performance on the CyberShake and Epigenomics workflows in terms of cost and makespan. In addition, the DA-AGSA technique has achieved a reliability of 0.99, 0.98 and 0.98 on the three workflows with 1000 tasks. As a future extension, a new hybrid metaheuristics optimization technique can be implemented for further enhancing workflow scheduling in a cloud platform.

Nomenclature

Parameters	Definition
C_{VMi}	Execution cost
P_{VMi}	Processing capacity of the VMs
ET_j^l	Execution time
$T_j = \{T_1, T_2, T_3, \dots T_J\} \forall \{1,2,3,..J\}$	Task
S_{T_j}	Size of a task
P_{var_l}	Processing capacity of VMs
DT_{jk}	Transfer time of data
D_{out}	Generated data
bw	Bandwidth between every VM
PT_j^l	Total processing time
F	Gravitational force
e	Number of edges inter-connected with every task T_j
$M_1 and\ M_2$	Mass of the 1st and 2nd particles
R^2	Distance between the two particles
$rand_j$	Random coefficient value
$v_i^d(t)$	Velocity
$x_i^d(t+1)$	Position
$fit_i(t)$	Fitness value of an agent i
w_1 and w_2	Adaptive weight functions

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

Acknowledgments

We thank Vasavi College of Engineering (Autonomous), Hyderabad for the support extended towards this work.

References

[1] B. Han, L. Luo, G. Sheng, G. Li, and X. Jiang, “Framework of random matrix theory for power system data mining in a non-Gaussian environment”, *IEEE Access*, Vol. 4, pp. 9969-9977, 2016.

[2] Y. Wang, Y. Guo, Z. Guo, T. Baker, and W. Liu, “CLOSURE: A cloud scientific workflow scheduling algorithm based on attack-defense game model”, *Future Generation Computer Systems*, Vol. 111, pp. 460-474, 2020.

[3] H. Djigal, J. Feng, J. Lu, and J. Ge, “IPPTS: an efficient algorithm for scientific workflow scheduling in heterogeneous computing systems”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 32, No. 5, pp. 1057-1071, 2020.

[4] F. Abazari, M. Analoui, H. Takabi, and S. Fu, “MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm”, *Simulation Modelling Practice and Theory*, Vol. 93, pp. 119-132, 2019.

[5] M. Karpagam, K. Geetha, and C. Rajan, “A reactive search optimization algorithm for scientific workflow scheduling using clustering

techniques", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 2, pp. 3199-3207, 2021.

[6] N. Wu, D. Zuo, and Z. Zhang, "Dynamic fault-tolerant workflow scheduling with hybrid spatial-temporal re-execution in clouds", *Information*, Vol. 10, No. 5, p. 169, 2019.

[7] A. Iranmanesh and H. R. Naji, "DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing", *Cluster Computing*, Vol. 24, No. 2, pp. 667-681, 2021.

[8] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning", *IEEE Access*, Vol. 7, pp. 39974-39982, 2019.

[9] M. Karpagam, K. Geetha, and C. Rajan, "A modified shuffled frog leaping algorithm for scientific workflow scheduling using clustering techniques", *Soft Computing*, Vol. 24, No. 1, pp. 637-646, 2020.

[10] 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, No. 2, pp. 1479-1503, 2021.

[11] A. Kaur, P. Singh, R. S. Batth, and C. P. Lim, "Deep-Q learning-based heterogeneous earliest finish time scheduling algorithm for scientific workflows in cloud", *Software: Practice and Experience*, Vol. 52, No. 3, pp. 689-709, 2022.

[12] 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.

[13] I. Gupta, M. S. Kumar, and P. K. Jana, "Efficient workflow scheduling algorithm for cloud computing system: a dynamic priority-based approach", *Arabian Journal for Science and Engineering*, Vol. 43, No. 12, pp. 7945-7960, 2018.

[14] S. Qin, D. Pi, Z. Shao, and Y. Xu, "Hybrid collaborative multi-objective fruit fly optimization algorithm for scheduling workflow in cloud environment", *Swarm and Evolutionary Computation*, Vol. 68, p. 101008, 2022.

[15] A. Aggarwal, P. Dimri, A. Agarwal, M. Verma, H. A. Alhumyani, and M. Masud, "IFFO: An Improved Fruit Fly Optimization Algorithm for Multiple Workflow Scheduling Minimizing Cost and Makespan in Cloud Computing Environments", *Mathematical Problems in Engineering*, Vol. 2021, p. 5205530, 2021.

[16] K. Loheswaran, "An upgraded fruit fly optimisation algorithm for solving task scheduling and resource management problem in cloud infrastructure", *IET Networks*, Vol. 10, No. 1, pp. 24-33, 2021.

[17] Y. Fu, M. Zhou, X. Guo, and L. Qi, "Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm", *Journal of Cleaner Production*, Vol. 278, pp. 123364, 2021.

[18] 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.

[19] L. Abualigah and A. Diabat, "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments", *Cluster Computing*, Vol. 24, No. 1, pp. 205-223, 2021.

[20] S. Saeedi, R. Khorsand, S. G. Bidgoli, and M. Ramezanpour, "Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing", *Computers & Industrial Engineering*, Vol. 147, p. 106649, 2020.

[21] J. K. Konjaang and L. Xu, "Multi-objective workflow optimization strategy (MOWOS) for cloud computing", *Journal of Cloud Computing*, Vol. 10, p. 11, 2021.

[22] M. Zeedan, G. Attiya, and N. E. Fishawy, "Enhanced hybrid multi-objective workflow scheduling approach based artificial bee colony in cloud computing", *Computing*, Vol. 105, No. 1, pp. 217-247, 2023.

[23] W. Li, J. Wu, Q. Zhang, K. Hu, and J. Li, "Trust-driven and QoS demand clustering analysis based cloud workflow scheduling strategies", *Cluster Computing*, Vol. 17, No. 3, pp. 1013-1030, 2014.

[24] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems", *IEEE Transactions on Vehicular Technology*, Vol. 68, No.10, pp. 10187-10200, 2019.

[25] A. Rezaeian, M. Naghibzadeh, and D. H. Epema, "Fair multiple-workflow scheduling with different quality-of-service goals", *The Journal of Supercomputing*, Vol. 75, No. 2, pp. 746-769, 2019.

[26] Z. G. Chen, Z. H. Zhan, Y. Lin, Y. J. Gong, T. L. Gu, F. Zhao, H. Q. Yuan, X. Chen, Q. Li, and J. Zhang, "Multiobjective cloud workflow scheduling: A multiple populations ant colony

Received: April 19, 2023. Revised: May 21, 2023.

system approach", *IEEE Transactions on Cybernetics*, Vol. 49, No. 8, pp. 2912-2926, 2019.

[27] J. Sun, L. Yin, M. Zou, Y. Zhang, T. Zhang, and J. Zhou, "Makespan-minimization workflow scheduling for complex networks with social groups in edge computing", *Journal of Systems Architecture*, Vol. 108, p. 101799, 2020.

[28] E. Rashedi, E. Rashedi, and H. N. Pour, "A comprehensive survey on gravitational search algorithm", *Swarm and Evolutionary Computation*, Vol. 41, pp. 141-158, 2018.

[29] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants", *Expert Systems with Applications*, Vol. 152, p. 113396, 2020.

[30] Y. Wang, S. Gao, M. Zhou, and Y. Yu, "A multi-layered gravitational search algorithm for function optimization and real-world problems", *IEEE/CAA Journal of Automatica Sinica*, Vol. 8, No. 1, pp. 94-109, 2020.