Int. j. inf. tecnol. (October 2024) 16(7):4419–4426
https://doi.org/10.1007/s41870-024-01881-3

ORIGINAL RESEARCH

Scientific workflow scheduling using adaptive dingo optimization in multi-cloud environment

A. Arul Mary[1]

Received: 8 January 2024 / Accepted: 10 April 2024 / Published online: 18 May 2024

Abstract A framework for emerging innovations and the capacity to provide reliable cloud services in cloud computing. The availability of "unlimited" computing capabilities to consumers on command is one of the key components of cloud computing. Single cloud holding resources, meanwhile, are typically constrained and could not be capable to handle the unexpected spike in user demands. To allow resource exchange amongst clouds, the multi-cloud architecture is proposed. Offering resources and activities across several clouds is a paradigm that is getting more and more popular today. The majority of existing cloud workflow scheduling projects focus on reducing costs or length of time. The greatest crucial Quality of Service (QoS) parameter, nevertheless, is the dependability of workflow scheduling. As a result, multi-objective scheduling for scientific processing in a multi-cloud architecture is suggested in this research to reduce workflow duration and expense while also satisfying the dependability requirement. To achieve this concept Adaptive Dingo Optimization (ADO) algorithm is designed. The proposed algorithm takes solution encoding, fitness calculation, and update functions. For experimental analysis, a different workflow model is used. The performance of the proposed approach is evaluated in terms of different metrics.

Keywords Quality of service · Adaptive dingo optimization algorithm · Makespan · Workflow scheduling · Multi-objective

✉ A. Arul Mary
arulmary1008@gmail.com

[1] Guest Lecturer, Department of Computer Science, Government Arts and Science College for women, Koothanallur, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India

1 Introduction

To provide communications, cloud computing has received enormous attention in recent years as well as it needs massive resources with services for executing large-scale applications [1]. Therefore, it joins similar ideas also knowledge toward giving mutual assets, tools, software, and information meant for individual computers with other machines [2]. Several computer frameworks are suggested for the enormous quantity of data storage as well as the computer needs of cloud computing [3]. Virtual Machine (VM) is one of the main software in cloud computing, it supports all windows in single software [4]. Also, through cloud infrastructure users can run large-scale workloads on VMs hosted. It improves application completion time and enables parallel processing of application tasks [5]. Utilizing accessible computing resources as effectively as possible is the primary goal of a cloud computing system. Scheduling tasks in suitable order so they can be completed under problem-specific limitations is the primary goal of scheduling [6].

Workflow scheduling problem on resources is the most essential issue for the uses of cloud environment [7]. Single cloud computing is flat to resource problems like hardware breakdown, software breakdown as well as power breakdown similar to other distributed computing [8]. Single cloud computing meets some disadvantages given user needs in scheduling tasks [9]. While operating a complex workflow program, these issues are expected and fixed develop task failures and workflow system faults [10]. Thus, the method considers the multi-cloud technique, it satisfies the customers with a variety of options and toward greatest assure their purpose necessities, particularly intended for persons scientific computing request [11]. Between various cloud contributors, this might be the better answer for the exchange of resources [12]. The cloud is a platform for large dispersed computing that is available for streaming for various process applications [13].

A crucial performance metric for cloud-based process scheduling is durability [14]. However, the reliability criteria used by various cloud providers vary. Users must therefore pay particular consideration to the dependability restriction of the workflow when choosing computer resources in a multi-cloud environment [15]. We have suggested multi-objective scheduling in this paper, which is taken into account for scientific processes in a multi-cloud context. The primary goal is to reduce makespan time and cost while also following dependability constraints. An optimization algorithm like the Adaptive Dingo Optimization Algorithm (ADO) is proposed to achieve the multi-objective scheduling with optimal values.

2 Related works

Many of the researchers had developed workflow scheduling on the cloud. Among them some of the works are analyzed here.

Hu et al. [16] had explained a multi-objective scheduling (MOS) algorithm for scientific workflow in a multi-cloud environment. The MOS algorithm is based on Particle Swarm Optimization (PSO), as well as the equivalent code strategy to get the tasks finishing place, as well as tasks, categorize information transmission toward concern. Widespread experimental explained the important multi-objective presentation development of MOS algorithm better result compared with CMOHEFT algorithm as well as the RANDOM algorithm.

Adhikari et al. [17] had discovered via allowing for numerous contradictory objectives utilizing the Firefly Algorithm (FA). The goal of FA is to match each workflow with a relevant cloud platform that can accommodate its requirements for load balancing and resource usage of the cloud servers. Additionally, a rule-based technique is built to distribute the tasks to the appropriate VM instances to shorten the duration of the workflow as well as gather the deadline. Using the approach, parameters like makespan, reliability, resource utilization and loads of the cloud servers were well defined.

For scheduling dependent tasks toward VMs Abed-Alguni et al. [18] introduce a discrete variation of the Distributed Grey wolf Optimizer (DGWO). The computation, as well as data transmission costs, are the two objectives for the scheduling process in DGWO. Compared with existing algorithms like PSO and GWO, DGWO gives better results and it was experimentally tested. Also, compared with other tested algorithms, the experimental outcome suggests that DGWO distributes tasks to VMs faster and it gives the makespan time. Based on Adaptive resource Allocation as well as Consolidation an Online Workflow Scheduling algorithm was proposed by Chen et al. [19]. This algorithm performs better compared to other methods.

Alaei et al. [20] develop an adaptive fault detector strategy based on the Improved Differential Evolution (IDE) algorithm. The study utilizes an Adaptive Network-Based Fuzzy Inference System (ANFIS) forecasting framework to prevent resource load fluctuation, which improves fault predictive performance before fault development. Compared with existing techniques, the method considerably enhances the whole arrangement presentation, attains a superior quantity of error acceptance with tall Hyper Volume (HV) contrasted by way of ICFWS, IDE, also ACO algorithms, reduces the makespan, the power expenditure as well as job error fraction, and minimizes the overall cost (The summary of literature survey is presented in Table 1).

Chakravarthi et al. [21] had proposed a scheduling algorithm called as Multi-Criteria Decision Making (MCDM) approach. According to the task requirement, a weighted total of run time cost, as well as the transfer of data instances, is utilized to determine the optimal resource among the available resources. The investigational outcome revealed that T-BDMWS provides present modern heuristics among the condition of attaining the user-specified resources otherwise deadline constriction and resource competence.

Medara et al. [22] presented an energy-aware algorithm in cloud computing called EASVMC with VM consolidation. It takes the multi-objective like consumption of energy, resource utilization as well as VM migrations. Algorithm-like inspired meta-heuristic approach called the Water Wave Optimization (WWO) toward lessening the energy consumption which finds an appropriate migration also lever off redundant hosts after migrating its VMs toward a suitable target swarm.

3 Workflow model using DAG

A scientific workflow is consist of several tasks which is modeled as a DAG. A workflow can be modeled as $W = (\mathrm{A}, E)$, where $\mathrm{A} = \{\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_n\}$ denotes the set of n tasks in the workflow as well as $E = \{(\mathrm{T}_i, \mathrm{T}_j) | \mathrm{T}_i, \mathrm{T}_j \in \mathrm{A}\}$ denotes the set of task dependencies. All tasks in the process must be completed within the deadline, which is a deadline constraint D attached to the workflow W. The simple representation of the DAG workflow [8] is shown in Fig. 1.

T1 and T10 represent the incoming and outgoing tasks, respectively, in Fig. 1. Tasks T2 and T3 are offspring of task T1. Only once the parent's job has been completed may the child's tasks be run. To be discovered is the ideal task–VM (task, VM) pair.

Table 1 Summary of literature survey

Authors	Objectives	Result	Limitations
Proposed Hu and Haiyang [16]	Makespan time, cost, and reliability constraints	Good makespan time, less cost, and better reliability	Time complexity is high due to the present VM
Adhikari et al. [17]	The workload of cloud servers, makespan time, resource utilization, and reliability	Minimum makespan and minimum cost	Time complexity is high
Zheyi et al. [18]	Resource utilization and total cost	Total execution of cost is minimized and solves the scheduling problem	Low convergence rate
Mani et al. [19]	Resource utilization and total cost of the scientific workflow model	Achieved average resource utilization and execution cost	During the execution, Only one task is executed at a time
Chakravarthiand Shyamala [20]	Energy consumption, total cost, and makespan time	Fault ratio is reduced in workflow scheduling	Low convergence rate
Chakravarthi et al. [21]	Run time, cost, and resource utilization	Minimum cost and efficient resource utilization	Not consider the reliability value
Elgendya et al. [22]	Energy consumption, resource utilization, and VM relocation	Reduced energy consumption	Doesn't operate two tasks in one VM

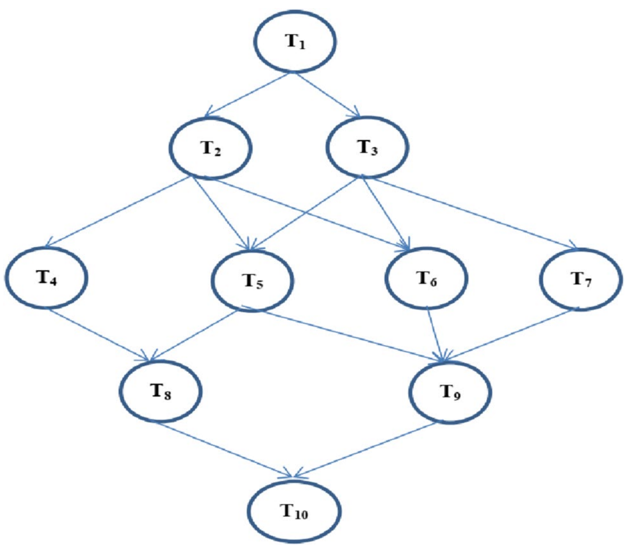

Fig. 1 Simple DAG model

4 Multi-objective function design

To run extensive experimental operations, an IaaS platform offers processing resources in the form of VMs. The proposed approach is taken into account in this scenario as a multi-cloud system for process scheduling. Two or more cloud processes connecting over the Internet make up a multi-cloud architecture. Cloud customers have the availability of VM resources from all cloud providers in our multi-cloud ecosystem. The features of each cloud, however, and the specific pricing structures of cloud platforms vary. A global cloud manager receives an application, divides the related workflow into numerous tasks, and presents them to the cloud scheduler. The tasks are scheduled by the cloud scheduler using a multi-objective scheduling technique. The tasks are then distributed to the accessible VMs using the local scheduler installed in each data center. The three characteristics of makespan, cost, and dependability were taken into account by the proposed method for multi-objective scheduling. The main goals of the suggested scheduling model are to decrease each task's makespan, cost, and dependability restriction. Control characteristics are established to outline this problem. The equation can be used to describe the issue (1).

$$\begin{aligned} &Minimize: O(S) = (makespan, \cos t) \\ &Subject\ to: R \geq r_{con}\ ; \quad r_{con} \in [r_{\min}, r_{\max}] \end{aligned} \tag{1}$$

where

$$Reliability = R = \Pi_{a_i \in A} r(a_i) \tag{2}$$

$$r(a_i) = \exp(-\xi \cdot T_{rent}(a_i, VM(n,m))); \quad \xi > 0 \tag{3}$$

$$Cost = C = \sum_{a_i \in A} C(a_i, VM(n,m)) \tag{4}$$

$$makespan = A_{end}(a_{exit}) \tag{5}$$

$$A_{end}(a_i) = A_{start}(a_i) + A_{recev}(a_i) + A_{exeu}(a_i, VM(n,m)) \tag{6}$$

$$A_{exeu}(a_i, VM(n,m)) = \frac{W(a_i)}{P(n,m)} \tag{7}$$

where $P(n, m)$ represent the processing capacity.

The objective function mentioned above will be minimized in this study utilizing the proposed approach. For that, the proposed method considered the adaptive dingo optimization (ADO) algorithm.

5 Proposed workflow scheduling methodology

A framework for emerging capabilities and the capacity to provide reliable cloud services in cloud computing. The availability of "unlimited" computing resources to consumers on-demand is one of the key components of cloud computing. Single cloud holding resources, nevertheless, are typically constrained and could not be sufficient to handle the unexpected spike in user demands. To allow resource sharing amongst clouds, the multi-cloud concept is proposed. Offering resources and services from several clouds is a paradigm that is getting more and more popular today. This research proposes a multi-objective scheduling method for scientific workflow in a multi-cloud environment, to concurrently reduce workflow duration and cost while satisfying the reliability requirement. To achieve this concept adaptive dingo optimization (ADO) algorithm is designed. The overall flow diagram of the proposed model is shown in Fig. 2.

The main aim of the proposed model is,

- To improve the efficiency of cloud scheduling, a large number of tasks are mapped onto cloud resources using a workflow model. Here DAG (directed acyclic graph) is used for the workflow model.
- To support resource sharing between the clouds, the multi-cloud concept is introduced here.

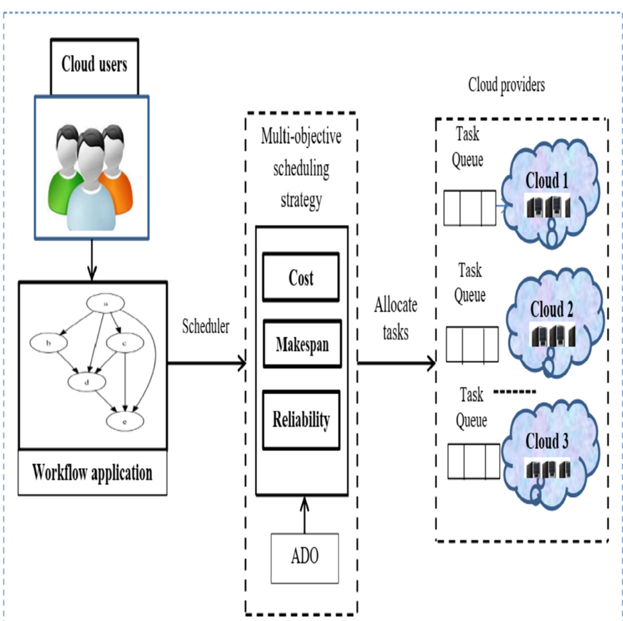

Fig. 2 Overall flow diagram of the proposed model

- Multi-objective function-based scheduling is presented using adaptive dingo optimization (ADO). This ADO helps to diminish the makespan and cost and improve the reliability constraint of the proposed workflow model.
- The efficiency of recommended technique is analyzed based on different metrics namely, makespan, cost, and reliability. A detailed explanation of each process is described in a further section.

5.1 Multi-objective scheduling strategy using ADO

The scheduling parameters such as makespan, cost, and reliability are optimized by using ADO in the scheduling process. The DOA is a brand-new bio-inspired global optimization algorithm that imitates dingoes' hunting tactics. Group hunting is an intriguing aspect of dingoes' social activity, which furthers the social behavior of dingoes. Predatory strategies are classified into their stages as follows: chasing and approaching, encircling and harassing, and attacking. Here the traditional dingo optimization is modified using updation process [23]. The step-by-step process of ADO is explained in beneath,

Step 1: Solution encoding

Solutions are representing the flow of tasks. In each optimization algorithm, encoding solution is a significant step. In this proposed work, the solution is considered as the task workflow. A workflow can be modeled as $W=(A, E)$, where A = {a1, a2, …, an} represents the set of n tasks in the workflow as well as $E=\{(ai, aj)|ai, aj \in A\}$ denotes the set of task dependencies.

Step 2: Fitness evaluation

Once the initial population is completed, the fitness function is computed. The fitness function is evaluated with the consideration of makespan, cost and reliability constrained. The objective of the proposed model is to minimize the makespan and cost value and improve the reliability constrained of each task. The fitness function is mathematically formulated in Eq. (1).

Step 3: Encircling process

After fitness evaluations, solutions are updated using dingo optimization. Dingoes are intelligent sufficient to locate their prey. The pack, led by the alpha, circles the prey after locating it. The following analytical solutions simulate these dingoes' behavior,

$$\vec{D}_d = \left|\vec{A} \cdot \vec{P}_p(X) - \vec{P}(i)\right| \tag{8}$$

$$\vec{P}(i+1) = \vec{P}_p(X) - \vec{B}.\vec{D}(d) \tag{9}$$

$$\vec{A} = 2.\vec{a}_1 \tag{10}$$

$$\vec{B} = 2\,\vec{b}\,.\,\underset{2}{\vec{a}} - \vec{b} \tag{11}$$

$$\vec{b} = 3 - \left(1 * \left(\frac{3}{l_{\max}}\right)\right) \tag{12}$$

positions of neighboring dingoes are addressed using a two-tier level vector given in Fig. 3. As mentioned by the location of the prey (P ∗, Q ∗), a dingo (P, Q) can update its position in place. By changing the value of the vectors $\vec{A}$ and $\vec{B}$ for the current area, each of the possible areas is arranged individually on the map around the best expert. The position vector of dingoes [23] is presented in Fig. 3.

Step 4: Hunting process

The position of possible prey is well known to all of the pack members, particularly alpha, beta, and others. The hunt is always led by the dominant dingo. Beta and other dingoes, though, could occasionally join in on the hunts. Other dingoes must change their positions by the location of the best search agent.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{a_1}.\overrightarrow{P_\alpha} - \vec{p}\right| \tag{13}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{a_1}.\overrightarrow{P_\beta} - \vec{p}\right| \tag{14}$$

$$\overrightarrow{D_0} = \left|\overrightarrow{a_1}.\overrightarrow{P_0} - \vec{p}\right| \tag{15}$$

$$\overrightarrow{P_1} = \left|\overrightarrow{P_\alpha}.\vec{b} - \overrightarrow{D_\alpha}\right| \tag{16}$$

$$\overrightarrow{P_2} = \left|\overrightarrow{P_\beta}.\vec{b} - \overrightarrow{D_\beta}\right| \tag{17}$$

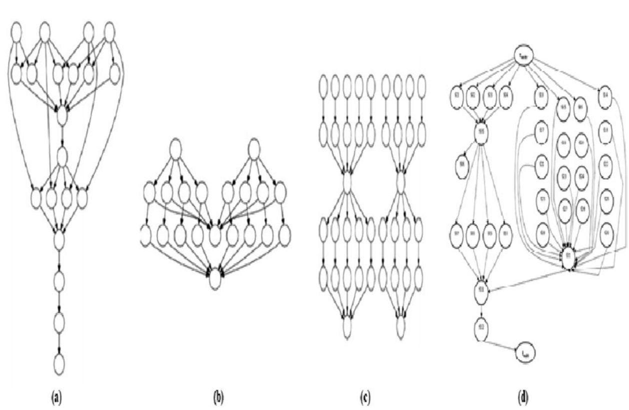

Fig. 3 Structure of workflows **a** Montage, **b** CyberShake, and **c** LIGO

$$\overrightarrow{P_3} = \left|\overrightarrow{P_0}.\vec{b} - \overrightarrow{D_0}\right| \tag{18}$$

In the dingo optimizer, the intensity of every dingo is computed based on the below equations,

$$\overrightarrow{I_\alpha} = log\left(\frac{1}{f_\alpha - (1E - 100)} + 1\right) \tag{19}$$

$$\overrightarrow{I_\beta} = log\left(\frac{1}{f_\beta - (1E - 100)} + 1\right) \tag{20}$$

$$\overrightarrow{I_0} = log\left(\frac{1}{f_0 - (1E - 100)} + 1\right) \tag{21}$$

In this, we can undoubtedly portray the situation updated by alpha, beta, and remaining dingo. Dingos (alpha, beta, and others) may also feel updated. Their positions are arbitrary and work in space prey in the hunting ground.

Step 5: Attacking prey process

Dingo concluded the hunt by assaulting the target if there is no location update.

Step 6: Searching process

Dingoes search for prey primarily based on the position of the pack. They continuously move forward to pursue and attack predators.

Step 7: Termination criteria

When the best fitness value is chosen, the algorithm stops working. The allocation procedure uses the chosen ideal task. The algorithm of scheduling is presented in Table 2.

6 Result and discussion

This section analyses the actual outcomes of the suggested workflow scheduling. Java is used to execute the suggested scheduling technique. The device is powered by an Intel Core i5 CPU and runs Windows 10 on a device with 6 GB of RAM. Java is used to simulate the suggested approach. Four different types of workflows, including Montage, CyberShake, Sipht, and LIGO, are employed for experimentation analysis [24, 25]. Figure 3, shows the workflow's organizational structure.

6.1 Experimental results

Here, the experimental analysis of various workflow models has been taken also simulation is done with the help of a cloud simulator in the Java domain to minimize the makespan time, and cost and satisfy the reliability constraint. The proposed algorithm is compared with existing algorithms

Table 2 Algorithm for ADO based workflow scheduling

Input: Scientific workflow model, Maximum number of iterations, numbers of population size, available **VMs**, available PM, total tasks, the parameter of ADO algorithm.
Output: Scheduled Task (Best solution)

Start
Randomly generate the initial solution
initialize the values
While termination conditions not reached do
Evaluate the fitness of each dingo's (solution)
Estimate the best search dingo (D_a)
Estimate the second-best search Dingo (D_b)
Estimate the remaining search Dingoes (D_o)
iteration 1
repeat
for i=1:D in do
Update the latest search agent status
end for
Evaluate the intensity cost and fitness of dingoes
Store the value of Da, D_b, D_o
Store the value of b, A, and B
Iteration=Iteration+1
check if, iteration> stopping criteria
output
End while

like FA, GWO, and PSO. The dataset details are given in Tables 3 and 4.

Table 5 explains the best score value for various tasks that correspond with virtual machines. For algorithm iterations, we have obtained distinct values. And it is calculated for our proposed and existing algorithms. In task T1, the best score obtained for ADO is 245, FA is 243and GWO is 241. Similarly, other algorithms give their best values. Our proposed model ADO gives the best score attained compared with existing algorithms. Also, we have gotten the fitness value best for the proposed ADO algorithm.

Figure 4, discusses to the minimum makespan time based on various iterations. Here the iterations are changed from 20 to 40. By utilizing the three algorithms the makespan time differs from 1200 to1550. The makespan time is contrasted with three algorithms FA, GWO, and ADO. We have acquired the greatest influence traverse to time got for FA and GWO algorithm. In our proposed ADO algorithm, we have accomplished the least makespan time.

Figure 5, explains a comparative chart for various tasks done with different scientific workflow models. The models are a montage, SIPHT, LIGO, and Cybershake. The X-axis

Table 3 Amazon EC2 VM instance specification

Instance type	"Core speed"	"Processing cores"	"RAM (GB)"	"Storage (GB)"	"Cost per hour" ($)
m1.small	1	1	1.6	150	0.5
m1.large	5	3	7.6	900	0.22
m1.xlarge	9	5	16	1710	0.51
c1.medium	6	3	1.8	370	0.29
c1.xlarge	21	9	7.2	1710	1.20

Table 4 Google compute engine VM instance specification

Instance type	"Core speed"	"Processing cores"	"RAM (GB)"	"Storage (GB)"	"Cost per hour" ($)
m1.small	1	1	1.9	160	0.7
m1.large	5	3	7.6	920	0.39
m1.xlarge	9	5	15.5	1720	0.52
c1.medium	6	3	1.9	350	0.4
c1.xlarge	21	9	7.5	1720	1.30

Table 5 Various tasks based on algorithm iterations

VM	Tasks	Iterations	Best score obtained			Fitness for proposed ADO
			ADO	FA	GWO	99
1	T1	10	245	243	241	98
2	T2	20	256	254	200	98.56
3	T3	30	265	260	262	99.21
4	T4	40	321	320	319	96
5	T5	50	342	340	300	97

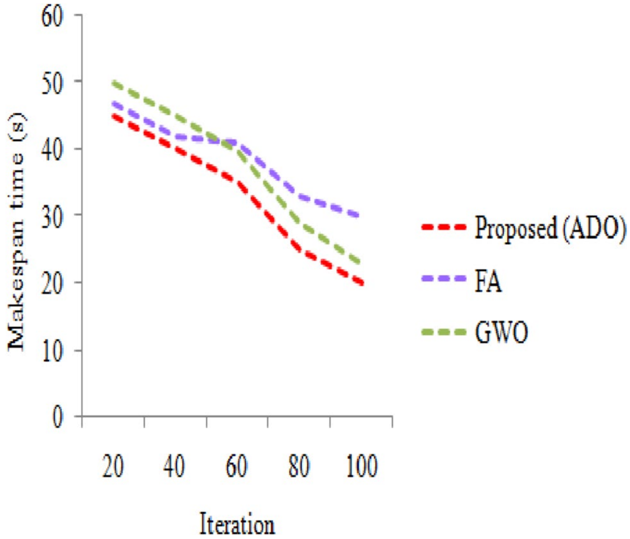

Fig. 4 Convergence graph for makespan time

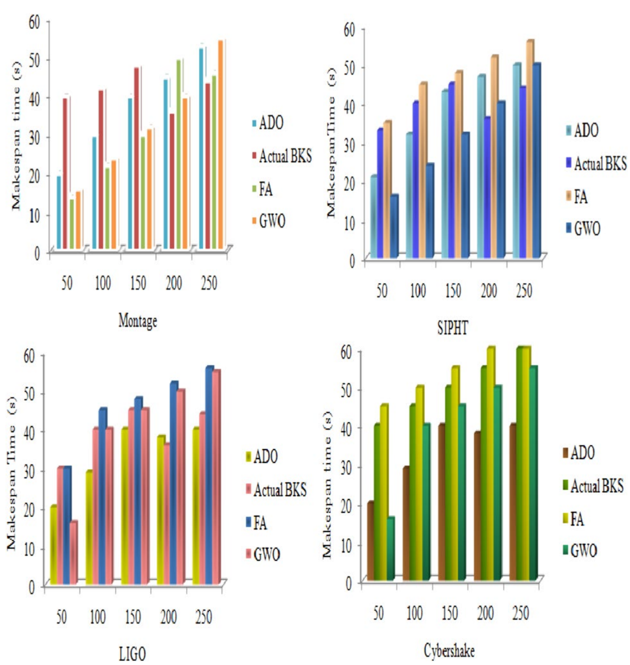

Fig. 5 Comparative analysis for time

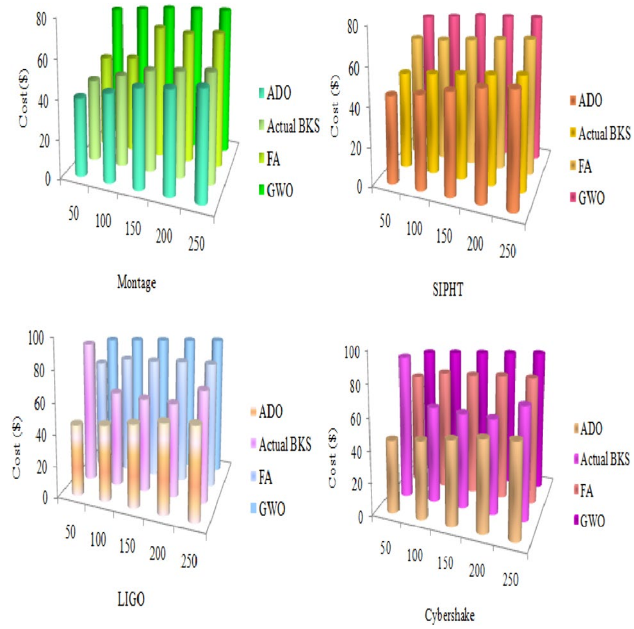

Fig. 6 Comparative analysis of Cost

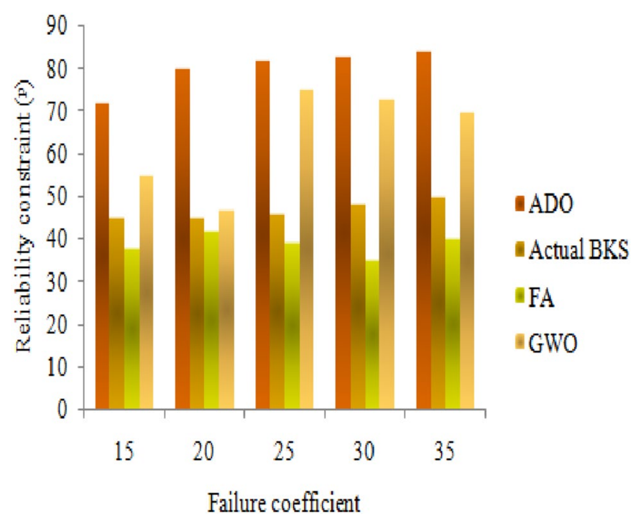

Fig. 7 Failure coefficient

mentions various tasks from the user and the y-axis mentions makespan time represented in h. Here, our existing algorithms like FA, GWO, and GSO. Here, we have gotten less time using our proposed ADO algorithm contrasted with all other algorithms.

Figure 6, explains the cost analysis compared with various optimization algorithms. Different workflow models have been taken for this analysis. We have achieved low cost in our proposed ADO algorithm. Compared with the existing algorithm our method gives the best solutions.

Figure 7, demonstrates the failure coefficient analysis for reliability constraint. In this, we have considered the failure coefficient 0.001, 0.002, and 0.003 compared with existing algorithms. The X-axis represents various tasks analyzed with failure coefficient and the y-axis represents the reliability constraint for our proposed and existing algorithms. Compared to existing techniques, ADO gives better reliability.

7 Conclusion

In this paper, we have proposed a multi-objective scheduling task taken for scientific workflow in a multi-cloud environment. Here, our main aim is to reduce the multi-objective parameters like reliability, makespan time, and cost. It is obtained by utilizing the fitness value of the algorithm our proposed algorithm. Further, the better execution of the ADO algorithm gives the least time, cost, and reliability as well as gives optimal value. The execution of the proposed scientific workflow scheduling strategy was analyzed and the test comes about to demonstrate that the proposed scheduling procedure has accomplished high exactness and effectiveness than the existing methods. Compared with existing algorithms like FA and GWO, our proposed algorithm gives better performance. In future work we will take on new algorithms on another platform also we will analyze faults that occurred during transmitting data which affects the data path.

Funding The authors declare that they have no Funding.

Availability of data and material Data sharing is applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Doostali S, Babamir SM, Eini M (2021) CP-PGWO: multi-objective workflow scheduling for cloud computing using critical path. Clust Comput 4:3607–3627
2. Sanaj MS, Prathap PMJ (2020) Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. Eng Sci Technol Int J 23(4):891–902
3. Jena RK (2015) Multi objective task scheduling in cloud environment using nested PSO framework. Proced Comput Sci 57:1219–1227
4. Abazari F, Analoui M, Takabi H, Song Fu (2019) MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm. Simul Model Pract Theory 93:119–132
5. Srichandan S, Kumar TA, Bibhudatta S (2018) Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. Future Comput Inf J 2:210–230
6. Jethava AN, Desai MR (2019) Optimizing multi objective based dynamic workflow using aco and black hole algorithm in cloud computing. In: 2019 3rd international conference on computing methodologies and communication (ICCMC). IEEE, pp 1144–1147
7. Heilig L, Buyya R, Voß S (2017) Location-aware brokering for consumers in multi-cloud computing environments. J Netw Comput Appl 95:79–93
8. Pasdar A, Lee YC, Almiani K (2018) Hybrid scheduling for scientific workflows on hybrid clouds. Comput Netw 181:107–438
9. Sooezi N, Abrishami S (2015) Scheduling data-driven workflows in multi-cloud environment. In: Proceedings of the 7th international conference on cloud computing technology and science (CloudCom). IEEE, pp 163–167
10. Kaur N, Singh S (2016) A budget-constrained time and reliability optimization bat algorithm for scheduling workflow applications in clouds. In: Shinde SV (ed), Proceedings of the 7th international conference on emerging ubiquitous systems and pervasive networks (EUSPN). Procedia Computer Science, London, pp 199–204
11. Hendre V (2023) Channel scheduling based interference lowering power efficient algorithm (CShILPeA) for the wireless body area network: design and performance analysis. Int J Inf Technol 15(1):169–182
12. Saleh E, Shastry C (2023) A new approach for global task scheduling in volunteer computing systems. Int J Inf Technol 15(1):239–247
13. Songara N, Jain MK (2023) MRA-VC: multiple resources aware virtual machine consolidation using particle swarm optimization. Int J Inf Technol 15(2):697–710
14. Godhrawala H, Sridaran R (2023) A dynamic Stackelberg game based multi-objective approach for effective resource allocation in cloud computing. Int J Inf Technol 15(2):803–818
15. Sumathi M, Vijayaraj N, Raja SP, Rajkamal M (2023) HHO-ACO hybridized load balancing technique in cloud computing. Int J Inf Technol 15(3):1357–1365
16. Li F, Seok MG, Cai W (2021) A new double rank-based multi-workflow scheduling with multi-objective optimization in cloud environments. In: IEEE international parallel and distributed processing symposium workshops (IPDPSW). IEEE, pp 36–45
17. Hu H, Li Z, Hua Hu, Chen J, Ge J, Li C, Chang V (2018) Multi-objective scheduling for scientific workflow in multicloud environment. J Netw Comput Appl 114:108–122
18. Adhikari M, Amgoth T, Srirama SN (2020) Multi-objective scheduling strategy for scientific workflows in cloud environment: a firefly-based approach. Appl Soft Comput 9:106–411
19. Abed-Alguni BH, Alawad NA (2021) Distributed Grey Wolf Optimizer for scheduling of workflow applications in cloud environments. Appl Soft Comput 102:107–113
20. Chen Z, Lin K, Lin B, Chen X, Zheng X, Rong C (2020) Adaptive resource allocation and consolidation for scientific workflow scheduling in multi-cloud environments. IEEE Access 8:190173–190183
21. Alaei M, Khorsand R, Ramezanpour M (2021) An adaptive fault detector strategy for scientific workflow scheduling based on improved differential evolution algorithm in cloud. Appl Soft Comput 99(2021):1–15
22. Chakravarthi KK, Shyamala L (2021) TOPSIS inspired budget and deadline aware multi-workflow scheduling for cloud computing. J Syst Archit 114:1–17
23. Bairwa AK, Joshi S, Singh D (2021) Dingo optimizer: a nature-inspired metaheuristic approach for engineering problems. Math Probl Eng 2021:1
24. Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G, Vahi K (2013) Characterizing and profiling scientific workflows. Future Gener Comput Syst 29:682–692
25. Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ, Mayani R, Chen W, da Silva RF, Livny M et al (2015) Pegasus, a workflow management system for science automation. Future Gener Comput Syst 46:17–35
26. Elgendy A, Yan J, Zhang M (2019) Integrated strategies to an improved genetic algorithm for allocating and scheduling multi-task in cloud manufacturing environment. Proced Manuf 39:1872–1879