Multimedia Tools and Applications (2024) 83:2527–2550
https://doi.org/10.1007/s11042-023-15687-1

DAGWO based secure task scheduling in Multi-Cloud environment with risk probability

Prashant Balkrishna Jawade[1] · **S. Ramachandram**[2]

Received: 15 December 2021 / Revised: 8 July 2022 / Accepted: 22 April 2023 /
Published online: 15 May 2023

Abstract
The Cloud with pay-per-use functioning has attained a great attraction towards on-demand applications. However, the availability of such services by a single data centre is limited, particularly, during the peak demand season. This is due to the fact that it has restricted resource availability. Therefore, the multi-cloud framework has been implemented. In this framework, more clouds get incorporated in a shared manner. All-private, public, or a blend of both may be a multi-cloud environment. The count of virtual machines is higher in the public cloud; however, security is not ensured. So far, most of the works have considered only the metrics like makespan, execution time, and execution time while allocating the tasks. However, the assurances of security while tasks' execution is still an issue in many complex environments. This research work intends to propose a secure task scheduling scheme in the multi-cloud environment with the assessment of risk probability. The suggested study focuses on using the optimization idea to allocate the tasks in the best way possible. As a result, the suggested optimal task allocation includes four objectives like "makespan, execution time, utilization cost, and security constraints (risk evaluation)". Also, a unique hybrid technique called Dragon Aided Grey Wolf optimization (DAGWO) is presented to address this optimization problem. Lastly, the performance of the suggested scheme is compared with theextantapproachesin terms of convergence, energy, makespan, etc. Especially, the risk probability of the proposed model while scheduling 100 tasks is 5.99%, 49.93%, 50.10%, 21.48%, and 31.557% better than existing PSO, WOA, DA, GWO, and MGWO methods respectively.

Keywords Multi-Cloud · Resource Allocation · Resource Scheduling · 4 fold Objectives · DAGWO

✉ Prashant Balkrishna Jawade
prashandjawade1234@gmail.com

1 Computer Science and Engineering, Government College of Engineering, Nagpur, India

2 Computer Science and Engineering, University College of Engineering, Osmaniya University, Hyderabad, India

Nomenclature

ADA	Adaptive Dragonfly Algorithm
ANN	Artificial Neural Network
CSP	Cloud Service Provider
CSSA	Chaotic Squirrel Search Algorithm
CSP	Cloud Service Provider
CU	Customer
DA	Dragonfly Algorithm
DAGWO	Dragon Aided Grey Wolf Optimization
EDA-GA	Estimation Of Distribution Algorithm And GA
ETC	Expected Time To Compute
FF	Firefly
GA	Genetic Algorithm
GWO	Grey Wolf Optimization Algorithm
MGWO	Modified Mean Grey Wolf Optimization Algorithm
MS	Make Span
NN-DNSGA-II	Neural Network-based Dynamic Non-dominated Sorting Genetic Algorithm
SLA-LB	Service Level Agreement-Based Load Balancing
TBTS	Threshold Based Task Scheduling Algorithm
TS	Task Scheduling
VM	Virtual Machines

1 Introduction

Cloud computing is a network-based evolving technology that aims to deliver different IT services for a wide range of enterprises and customers on a pay-for-use basis [26–28]. Cloud computing offers many advantages with respect to low cost and data usability. Especially, cloud computing promotes improvedresponsiveness to supply chain interruptions [41]. Furthermore, the cloud model is used in review-based recommender systems [3], and biometric authentication [19]. In the cloud, data values from data owners are handled by the cloud provider [6, 13, 55]. Moreover, the risks of service degradation and the malicious insider are the issues in the single cloud. However, the multi-cloud environment monitors several cloud infrastructures and prevents dependence on other clouds [8, 9, 34, 46]. The storage and device partitioning mechanism [37] is used as the multi-cloud resource sharing scheme. For the delivery of services to various cloud vendors, four types of architectures are used. They are application duplication, application device partition, application logic partition in fragments, and application data partition in fragments [2, 3].

In the cloud, the data centre handles huge requests for applications from every corner of the world [17, 38]. In each application, there have many contingent and autonomous activities. To execute such requirements as well as to decide the execution order of the tasks, each cloud needs a scheduling strategy. Different clouds may get their scheduling strategies [5, 44, 51]. Task scheduling objectives [1, 11] specifically involve reducing the time and energy consumption of task execution and optimizing the usage of resources and the ability to balance the loads. Further, reducing work completion time is beneficial for enhancing the customer experience with the drastic rise of the number of cloud users [21] [7]. In order to avoid the performance degradation due to the enlargement of resources or waste obligatedvia unnecessary idle resources, load balancing capability gets improved, which further leads to maximum utilization of VM. The TS algorithm has therefore been found to be NP-complete, and it is not feasible to achieve the optimum solution in a finite amount of time [23, 49].

Till date, evolutionary algorithms such as GA and distribution estimation algorithms have been introduced to solve multiple scheduling and mapping problems [42, 48, 53, 54]. With the single-objective approach, schemes like "Min-Min, Max-Min, and Suffrage models" are implemented. However, they are not very extensible or adaptable. In most circumstances, the model merely takes into account the shortest possible time to complete the task, ignoring safety precautions.

The maincontribution of this research study is listed beneath:

- Asecure task scheduling scheme in the multi-cloud environment is proposed with the assessment of risk probability.
- For scheduling the tasks, a four-fold multi-objective constraint-based optimization model is introduced.
- The defined 4 fold-objective constraints are "makespan, execution time, utilization cost, and risk probability". In addition, the weight in the objective function is calculated by the fuzzy triangular membership function.
- A unique hybrid DAGWO algorithm is proposed for optimal task scheduling. The proposed DAGWO model is a combination of traditional DA and GWO algorithms.

Paper organization Section 2 shows therecent studies in the same field. Section 3 discusses theoverview and problem statement of thesuggested secure task scheduling. Section 4 shows the cloud setup as well as defined 4 foldobjectives. Section 5 explaines the obtained outcomes. Section 6 ends the work.

2 Literature review

In this section, the latest task scheduling researches are catagorized based on their approaches.

2.1 Related works

a) ***Optimization based approaches***

In 2019, Pang *et al*. [37] have developed an EDA-GA-based hybrid algorithm for task scheduling. At first, EDA's sampling & probability concepts have beendeployed to givespecific workable solutions. The ideal scheduling model for assigning the work to VM's was finally realized.

In 2019, Sanaj and Joe [45] have explored CSSA for optimal "multitask scheduling in an IaaS cloud environment". The method continuously generates work schedules, which increases cost-effectiveness. The prior network was prouced using chaotic optimization for task allocation in order to assure superior global convergence.

In 2019, Yusuf [30] has introduced a new TS model, where the major intention relied on optimizing the TS by reducing the energy consumption and MS. The MGWO was able to solve the scheduling difficulties, which enhanced system efficiency. Encircling & hunting in MGWO have been modified utilizing the mean value to raise GWO's effectiveness.

In 2020, Neelima and Reddy [31] have developed a new load balancing task scheduling model in the cloud by means of ADA, which was an amalgamation of DA and FF algorithms. The multi-objective function was defined on the basis of"load, processing costs, & completion time" constraints toattain the enhanced performance.

In 2020, Ismayilov and Haluk [14] have investigated a prediction-oriented approach named as "NN-DNSGA-II algorithm" that incorporated NSGA-II with ANN framework. To address the job scheduling issue, the top five non-prediction-oriented dynamic techniques have beenutilized. The six objectives of the work that is being provided are "improvement of utilization and reliability, decrease in energy cost, makespan, & imbalance level".

b) ***other approaches***

In 2017, Liu *et al*. [22] have established a multi-task scheduling model, which incorporated a task workload model by considering service quantity and service coefficient. The effects of several workload-oriented task scheduling strategies have been then examined with respect to utilization & overall completion time.

In 2019, Panda *et al*. [36] have developed the multi-cloud network, where several clouds were integrated mutually for providing a combined service. In this method, an "allocation-aware task scheduling algorithm" was proposedand itdepend on the conventional "Min-Min and Max-Min algorithm".

In 2020, Lavanya *et al*. [20] have explored two allocation models termed TBTS and SLA-LB models. Task scheduling in VM with various setups is made easier by TBTS, which scheduled the tasks in batches. Tasks were dynamically scheduled by SLA-LB based on user requirements such as budget as well as deadline.

2.2 Research Gaps

Table 1 represents the review of recent works. The following are the limitations of the extant models.

(i) The SLA-LB model provides minimal complexity. But the main disadvantage was that the energy utilization factor was not evaluated [20]. Because the cost of energy utilization in the cloud is an important factor in determining the system's effectiveness.
(ii) In task scheduling, the continuous arrival of tasks is considered for effective system performance, but this factor is not considered in [22].
(iii) In the ADA model, VM scheduling was not clearly considered and it needs more consideration [31]. Furthermore, the MGWO model requires attention to task priority [30].
(iv) Security is one of the major issues in cloud computing. The related works failed to analyze the security constraints.

To overcome these limitations, a novel DAGWO based task scheduling is proposed in this study. Compared with the related works, this research introduces 4-fold objectives like"makespan, execution time, and utilization cost, along with security constraints". Moreover, a novel DAGWO model is utilized for optimal task scheduling.

Table 1 Advantages and difficulties of the extant task scheduling approaches

Author	Approach	Advantages	Difficulties
"Panda et al. [36]	Min-Min & Max-Min	• Lower complexity • Large utilization of cloud	• No consideration on fault tolerance.
Lavanya et al. [20]	SLA-LB	• Low execution time • Minimized penalty cost	• Energy utilization is not taken into account.
Pang et al. [37]	EDA-GA	• Enhanced load balance • Low execution time	• Require consideration on real cloud environments.
Liu et al. [22]	SAW	• Minimizedmake span • Greatly reliable	• The continuous arrival of tasks is not focused.
Sanaj and Joe [45]	CSSA	• Cost effective • Good throughput	• Analysis on real time has to be considered.
Ismayilov and Haluk [14]	NN	• Reliable • Minimizedmake span	• Needs concern on machine learning models.
Neelima and Reddy [31]	ADA	• Low execution cost • Enhanced load balancing	• Needsmore concern on VM scheduling.
Yusuf [30]	MGWO	• Low energy utilization • Minimizedmake span	• Have to focus more on task priority."

3 Proposed secure task scheduling model: an overview and problem statement

3.1 An Overview

The major components of the suggested work are listed below:

- Customer: This is the cloud's service client. The CU can produce their demands with the assistance of the cloud manager.
- Cloud Manager: This is a central entity that handles customers' service requests and receives the status of cloud providers' VMs.
- Cloud Service Provider: This is a cloud distributor that offers on-demand services by setting up virtual machines on actual servers. Each cloud contains a management server that contacts with other manager servers to send customer requests to manage the peak demands. Customers' requests are frequently scattered over several clouds as well as are handled using a distributed model.

Between the components of the cloud model, the following connections occur: "(a) CU-cloud manager, (b) cloud manager-CSP, (c) CSP-CSP, (d) CSP-cloud manager, and (e) cloud manager-CU". Furthermore, the overhead will happen to decide on the individual part. Nevertheless, we believe that the flexibility of the cloud model makes these overheads marginal.

Figure 1 represents a multi-cloud task scheduling scheme. The proposed model aims to produce satisfaction to the customer via effective task scheduling. Initially, users at various fields send a request tothe cloud. The cloud manager manages the service request and receives the cloud provider's VM status. If a task is submitted to the cloud manager, the manager adds this task to a waiting queue as well as locates an active VM to set the task to based on the specified 4-fold objectives. " Execution time, makespan, utilization cost, and risk probability" are the established 4-fold objectives in this research work. The work that has been presented deals with applying the optimization notion to task allocation. In the VMs, the tasks are continuously assigned, and the scheduling happens concurrently as well. A new hybrid method called DAGWO is presented for optimal scheduling. Furthermore, taking risk probability into account is the main contribution because this parameter guarantees the need for security while scheduling the activity. Each task's VM allocation is done based on the risk probability (and with other metrics). Finally, the efficient task is allocated with minimum execution time, minimum makespan, minimum utilization cost, and high security (low-risk probability).

4 Cloud setup and defined 4 fold-objective: makespan, execution time, utilization cost, risk probability

4.1 Cloud Setup

A set of clouds $c = c_1, c_2, c_3, \ldots, c_M$ is considered in this work. The cloud service provider CSP_I is denoted as c_I, $1 \leq I \leq M$. Here, the count of clouds, $M = 3$. The number of VMs ($|v_I|$), as well as the scheduling strategy, (S_I) will be assigned by the CSP_I. The CSP provides the

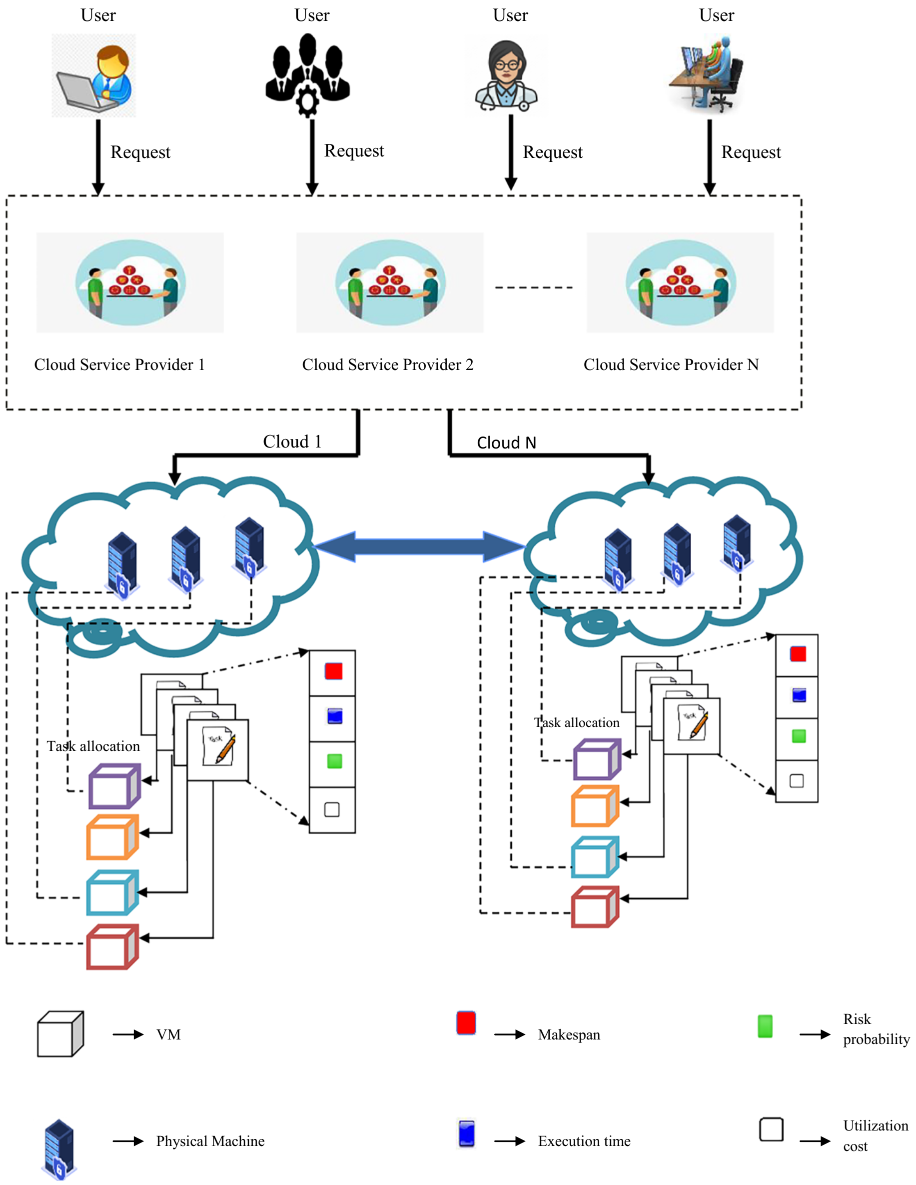


Fig. 1 The framework of the suggested task scheduling approach

clouds or processing certain requests from the user. Within the cloud, resides the PM with a huge number of VMs. The 3-tuple fashion is utilized to represent c_I as $c_I = <CSP_I, S_I, |v_I|>$, $1 \leq I \leq M$.

The request from the users is considered as the task. The task set $t = t_1, t_2, t_3, \ldots, t_L$ is of independent tasks in which t_I, $1 \leq I \leq L$ are I^{th} tasks with an instruction set ins_I (in MI). To process these tasks, a set of VM is considered $v = v_1, v_2, v_3, \ldots, v_M$, in which the VMs set is v_I, $1 \le I \le M$ as well as it is deployed under c_I. In addition, a virtual machine $VM_J \in v_I$; $1 \le I \le M$; $1 \le J \le |v_I|$ has the processing speed in MIPS P_J. The total VM in all the clouds is depicted by $M' = \sum_{I=1}^{m} | v_I |$.

The execution time of a task (t_I, $1 \le I \le L$) on a VM (VM_J, $1 \le J \le M'$) is given in the form of the matrix & it is reffered as ETC matrix. Mathematically, the ETC matrix is shown in Eq. (1) [36].

$$ETC = \begin{cases} & t_1 & t_2 & \cdots & t_L \\ VM_1 & ETC_{11} & ETC_{21} & \cdots & ETC_{11} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VM_{|v_1|} & ETC_{1|v_1|} & ETC_{11|v_1|} & \cdots & ETC_{L|v_1|} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VM_{\gamma+1} & ETC_{1(\gamma+1)} & ETC_{2(\gamma+1)} & \cdots & ETC_{1(\gamma+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VM_{\gamma+1} \mid v_M \mid & ETC_{1(\gamma+|v_M|)} & ETC_{2(\gamma+|v_M|)} & \cdots & ETC_{L(\gamma+|v_M|)} \end{cases} \tag{1}$$

In which, $\gamma = \sum_{K=1}^{M-1} | v_K |$. Moreover, ETC_{IJ}, $1 \le I \le L$; $1 \le J \le M'$ is the ratio of the instruction set (in MI) to the processing speed. Mathematically, ETC_{IJ} is given in Eq. (2).

$$ETC_{IJ} = \frac{ins_I}{P_J} \tag{2}$$

"The mapping is done in terms of allocating, matching, and scheduling of tasks." The mapping function Fgets the request set $r = \{r_1, r_2, r_3, \ldots r_R\}$ from the CUset, $CU = \{CU_1, CU_2, CU_3, \ldots CU_R\}$. The request $r_i = \{t_1, t_2, t_3, \ldots t_{L'}\}$; $1 \le I \le R$; $L' << L$ is assigned to the set of VM's $v = \{v_1, v_2, v_3, .., v_M\}$. Moreover, $|t| = \sum_{K=1}^{R} |r_i|$.

The secure mapping of the task onto the cloud, which has the VM set, $v(F : t \rightarrow v)$. This work intends to schedule the task optimally concerning the described 4-fold objective "(a) minimized makespan S, (b) lower cloud utilization cost of tasks U, (c) lower execution time of VME, (d) lower security risk probability G." The upcoming section comprehensively portrays these defined 4-fold objectives. The flow chart representation is shown in Fig. 2.

4.2 Defined 4-fold objectives

The objective of this study is evaluated as per Eq. (3). This Eq. (3) is also known as a fitness function and it is defined as: "A fitness value is a form of the objective function which summarises, in a single measure, how close a given design solution is to attaining the defined goals".

$$obj = \min \left\{ \left(W_1.S\right) + \left(W_2.U\right) + \left(W_3.E^{task}\right) + \left(W_4.G\right) \right\} \tag{3}$$

In which, S,U,E^{task}, and G denotes makespan, utilization cost of task, execution time for all tasks, and security risk probability. The weights W_1, W_2,W_3, & W_4 are calculated using fuzzy triangular membership function. The weight calculation is described in Eq. (4). Here p, q, r is the vertices of triangular membership function T(f). Lower boundaryis p, medium

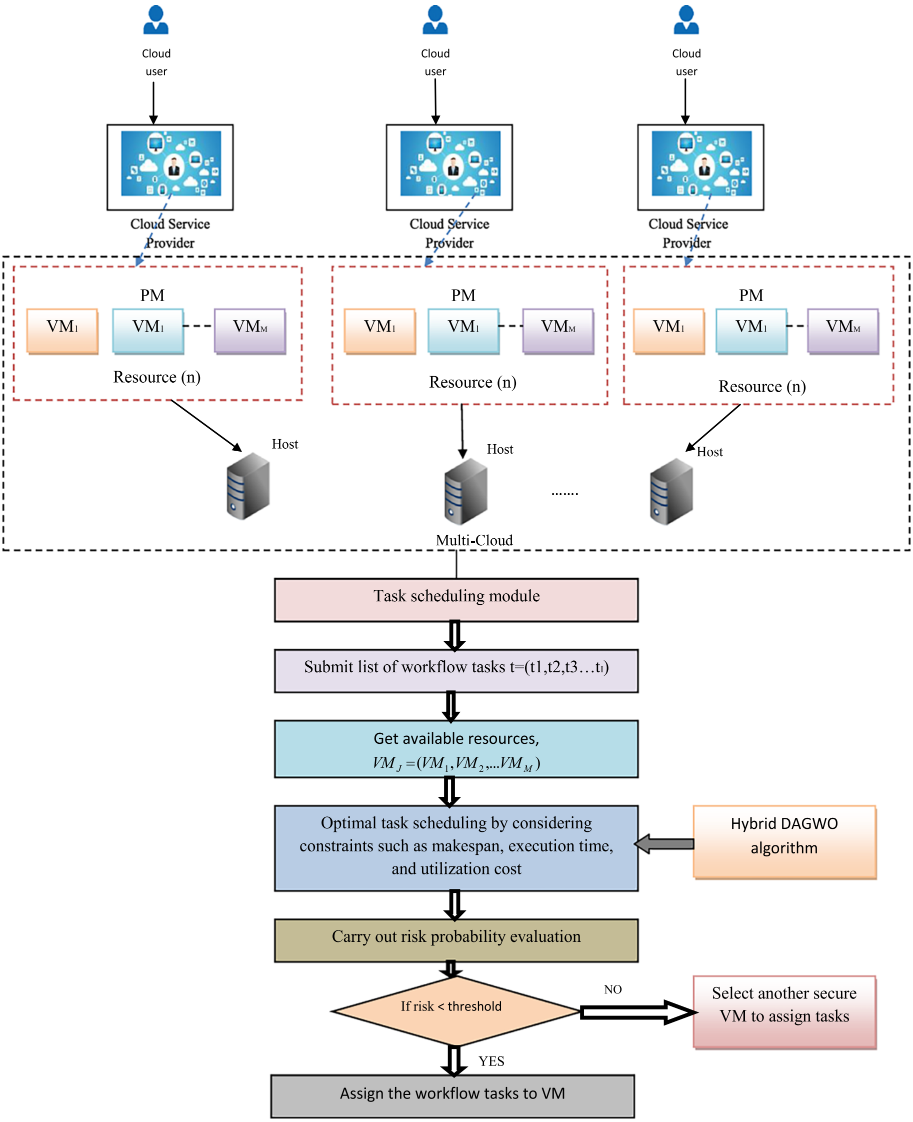


Fig. 2 Workflow model

boundary with membership value 1 is q as well as r is the upper boundary with membership value 0.

$$W = \begin{Bmatrix} 0 & ;if\ r < f \\ \frac{r-f}{p-f} & ;if\ f \le r \le p \\ \frac{q-r}{q-p} & ;if\ p \le r \le q \\ 0 & ;if\ r \ge q \end{Bmatrix} \tag{4}$$

Execution time of VM (E^{VM}) "Execution time of VM is the amount of time taken required by task to complete its executionon VM". The mathematical formula E^{VM} is given as per Eq. (5). In which, $H_{length(I,J)}$ is the task required to execute the instruction length, $V_{CPU(I,J)}$ is the count of CPU virtual machine J, and $V_{MIPS(I,J)}$ is the VM processing capability as VM J [10].

$$E^{VM} = \frac{H_{length\ (I,J)}}{V_{CPU(I,J)} \times V_{MIPS\ (I,J)}} \tag{5}$$

Utilization cost of tasks, U "Utilization cost is cost or total amount of payment from a cloud user to cloud provider against the utilization of resources to execute tasks". It is represented in Eq. (6). In which, $V_I$denotes the count of VM &ct_I indicates the completion time of VM. The expression for ct_I is given in Eq. (7), where, ***pesnumber*** is the count of processing element for running a task on a suitable VM, ***MIPS*** is the execution speed per processing element of a VM, and R is the count of tasks [24].

$$U_{task} = \sum_{I=1}^{V_I} PRICE_I * ct_I \tag{6}$$

$$ct_I = \sum_{J=1}^{R} \frac{H_J.LENGTH}{V_I.pesnumber \times V_I.MIPS}; I \in \{1, 2, \ldots . V_I\}\ and\ \ J \in \{1, 2, \ldots . R\} \tag{7}$$

Execution time for all tasks E^{task} "The amount of time needed to complete the execution of all tasks is referred as execution time for all tasks" and it is evaluated as per Eq. (8). Here, R is the count of tasks [32].

$$E^{task} = \frac{1}{\text{MAX (execution time)} \times R} \sum_{\text{I=1}}^{\text{R}} \text{(Execution time of respective VM} \times \text{Size of the task)} \tag{8}$$

Makespan S "Makespan is the cumulative time that the resources are required to complete the execution of all tasks". In general, VM usage is characterized by how well the resources in the cloud are used [35]. The scheduler is delivering effective and good task planning to resources if the makespan is low. The mathematical formula for S is given as per Eq. (9). Here, $V_I$denotes the number of VM and ct_I indicates the completion time of VM.

$$S = \underset{1 \le I \le V_I}{\text{MAX}} \left\{ ct_I \right\} \tag{9}$$

Security risk probability "The determination of the chance of a risk occurring is known as risk probability". It is used to measure the security risk in execution that is to evaluate the risk of the scheduling tasks. For a particular task, the scheduling risk probability of tasks $t_I VM_j$ are computed using Eq. (10).

$$\Pr ob^q\left(t_I^q, VM_I^q\right) = \begin{cases} 0 & if\ SD_I^q \leq SS_J^q \\ 1 - e^{-\left(SD_I^q \leq SS_J^q\right)} & otherwise \end{cases} \tag{10}$$

Here, the notation SD_I^q, SS_J^q denotes the security demand & security services t_I. "The risk probability of the task is the overall risk probability of the task corresponding to the security service." Fort_I, if $SD_I^q < SS_J^q$, then the risk probability of scheduling t_I on VM_J is zero. In addition, $Prob(t_I, VM_J)$ denotes the probability of t_I that is attacked during the execution and it is shown in Eq. (11).

$$\Pr ob\left(t_I, VM_J\right) = 1 - \prod_q \left(1 - \Pr ob^q\left(t_I^q, VM_J^q\right)\right) \tag{11}$$

In addition, the risk probability of workflow is an average of the probabilities of all tasks. The composed task's average probability $Prob(W)$ is being assaulted during the workflow in Eq. (12).

$$\Pr ob(W) = \frac{\sum_{t_I \in t} \Pr ob\left(t_I, VM_J\right)}{M} \tag{12}$$

The VM, as well as tasks, are the input solution to DAGWO for optimal scheduling with the consideration of the above-defined objectives. Since 3 PM with 30 sets of VM in each PM is utilized in this research study, the solution to the proposed model looks as in Fig. 3. In Fig. 3, the solution encoding of cloud1, cloud 2,& cloud 3 is manifested.

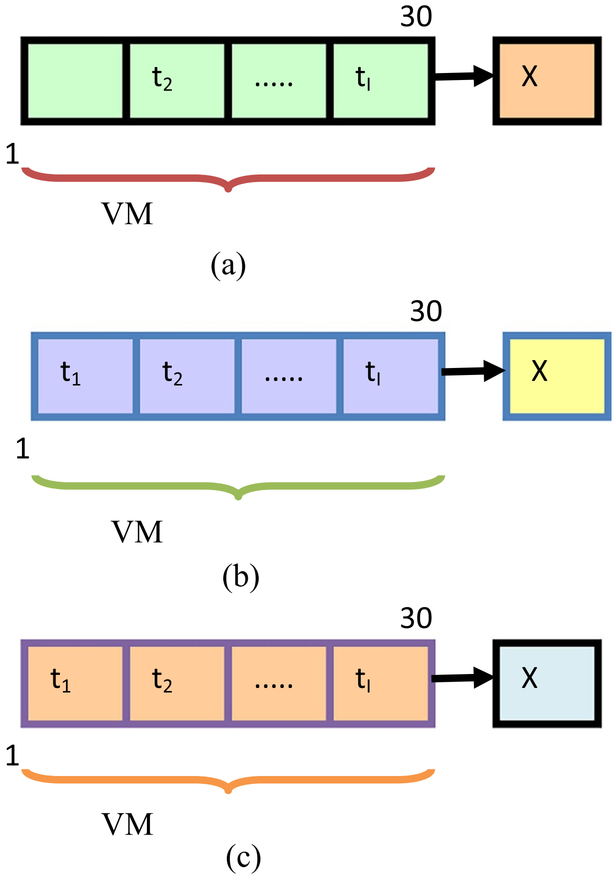
4.3 Proposed DAGWO for optimal task scheduling

Dragonfly Algorithm (DA) The DA [15, 16] was developed based on the inspiration of dragonflies. The three elementary principles like separation, alignment, and cohesion are observed by the swarming behavior of the dragonflies. In addition, each dragonfly should obey the separation operation, alignment operation, food attraction operation, cohesion operation, and enemy distraction operation.

Grey Wolf Optimization (GWO) Grey wolves are highly sociable animals with a very rigid hierarchy. They spend the majority of their lives hunting, seeking for prey first, encircling prey, as well as eventually attacking prey. Grey wolves have four primary societal structures: alpha, beta, delta, & omega, each of which plays a particular role in the group. The GWO algorithm is a new meta-heuristic based on these grey wolf characteristics [50] [47] [12, 52].

The DA has the benefit ofa higher convergence rate, which includes the ability to solve continuous problems. Similarly, the hunting habits of grey wolves served as an influence for the development of GWO. Since it has a straightforward structure, it is simple to build, requires less data than other techniques, converges quickly, and avoids local optima when

Fig. 3 Solution encoding (**a**) cloud 1, (**b**) cloud 2 & (**c**) cloud 3



used with composite functions. This research work intends to hybridize the concept of DA and GWO to make the model even stronger for optimal solutions with better convergence. Generally, hybridization of optimization algorithms can solve diverse classification and optimization problems [2, 47] [16, 25, 33, 40] [4, 39, 43]. The proposed model is referred to as DAGWO, and its step-by-step procedure is described below:

In the beginning, the population of the search agent is initialized as $X_i = X_1, X_2, \ldots, X_W$. The step vector ΔX_{iter+1} in the DA algorithm is deployed to evaluate the individual's movement direction and the mathematical formula is shown in Eq. (13). Here, the current iteration is denoted as ***iter***. Also, p, b, a, f, ***and*** e represents the separation weight, alignment weight, cohesion weight, food factor, as well as enemy factor. The inertia weight of the step vector is indicated by w.

$$\Delta X_{iter+1} = \left(pO_i + bB_i + aA_i + f.food_i + e.enemy_i\right) + w\Delta X_{iter} \tag{13}$$

Compute the value of separation criteria O, alignmen tB, cohesion criteria C, attraction to food resource (***food***) as well asdistractionaway froman enemy (***enemy***) by Eq. (14)-Eq. (18), respectievly. In Eq. (17) and Eq. (18), X^{+},X^{-}&Xsignifies the food sourceposition, enemy source, &position of the present individual.

$$O_i = \sum_{j=1}^{D} X - X_j \tag{14}$$

$$B_i = \frac{\sum_{j=1}^{D} X_j}{D} \tag{15}$$

$$A_i = \frac{\sum_{j=1}^{D} X_j}{D} - X \tag{16}$$

$$\boldsymbol{food}_i = X^+ - X \tag{17}$$

$$\boldsymbol{enemy}_i = X^- + X \tag{18}$$

The adaptive knowledge rate of i^{th} dragonfliesat the current iteration is computed by using Eq. (19).

$$d_i^{\boldsymbol{iter}} = \frac{1}{1 + e^{-V}} \tag{19}$$

Here,$V = \frac{|fit(X_i^{iter}) - fit(X_{best}^{iter})|}{fit(X_{best}^{iter}) + \kappa}$,$fit\left(X_i^{\boldsymbol{iter}}\right)$ is the fitness of the i^{th} search agent at $\boldsymbol{iter}^{th}$ iteration and $fit\left(X_{\boldsymbol{best}}^{\boldsymbol{iter}}\right)$ is the best fitness value. Also, κdenotes the constant value, which is employed to avoid the zero division error. Then update the radius of the neighbour. If there is only one neighbour to the present search agent, then update the search agent position by Eq. (20).

$$X_{iter+1}^{i} = d_{iter}^{i}.X_{iter}^{i} + \Delta XX_{iter+1}^{i} \tag{20}$$

Based on the hybrid proposed contribution, initialize a random variable *rand*. On the basis of this *rand*, the position update gets varied. The threshold value fixed here is 0.5, and if the assigned *rand* is less than the defined threshold value (*rand*<0.5), then update the search agent position utilizing standard GWO, else keep the existing solution as it is. The position update of GWO is evaluated as per Eq. (21).

$$X(\boldsymbol{iter} + 1) = \frac{X_1 + X_2 + X_3}{3} \tag{21}$$

In which, $X_1 = X_\alpha - Y_1.\left(s_\alpha\right),$
$X_2 = X_\beta - Y_2.\left(s_\beta\right),$
$X_3 = X_\delta - Y_3.\left(s_\delta\right).$

X_α, X_β & X_δ points to the position of the 1st, 2nd' and 3rd best solutions.

(Here,) $s_\alpha = | h_1.X_\alpha - X |,$
$s_\beta = | h_2.X_\beta - X |,$
$s_\delta = | h_3.X_\delta - X |$

In which, s, Y are the coefficient vectors and are mathematically defined in Eq. (22) and Eq. (23), respectively.

$$TY = 2e.u_1 - \vec{e} \tag{22}$$

$$h = 2.u_2 \tag{23}$$

Here, e is a constant, which is gradually reduced from 2 to 0 over the count of iterations. In addition, u_1 and u_2 is a random vector within 0 to 1.

Algorithm 1 manifests the pseudo-code of the DAGWO algorithm. Figure 4 represents the block diagram of the proposed approach.

5 Results & Discussions

5.1 Simulation procedure

The suggestedapproach was executed in **Python**. Three clouds as well as three sets of PM were used to make up the simulation scenario. The number of VMs in each group of PM is 30. The whole number of tasks to be completed ranges from 100 to 150 to 175 to 200, correspondingly. DAGWO outperforms other models in terms of optimal scheduling in terms of "migration cost, total cost, energy consumption, response time, and security analysis". The number of VMs used in this evaluation varies. Two scenarios are considered for evaluation. In scenario 1: 30 counts of VM are considered for scheduling the varying count of tasks (i.e.10 VM in each PM). In Scenario 2: 60 counts of VM are considered (i.e. 20

Algorithm 1 Pseudo code of DAGWO Algorithm

Algorithm 1: Pseudo code of DAGWO Algorithm
Initialize the population $X_i = X_1, X_2, ..., X_W$
step vector ΔX_{iter+1} is computed using Eq. (12)
While $iter < iter^{Max}$
Evaluate the search agent fitnessby Eq. (3).
Update $food$ and $enemy$ using Eq. (17) and Eq. (18), respectively.
Update the value of p, b, a, f, e and w of the i^{th} individual.
Compute the value of O, B, C, $food$ and $enemy$ using Eq. (14)- Eq. (18), respectievly
Compute the adaptive knowledge rate of i^{th} dragonfly $iter$ by Eq. (18)
Update the neighbour radius
Positional update via Eq. (20)
Initialize a random value $rand$
If $rand$ <0.5
Positional update via Eq. (21), the GWO update
else
Keep the existing solution
End if
end while
Terminate

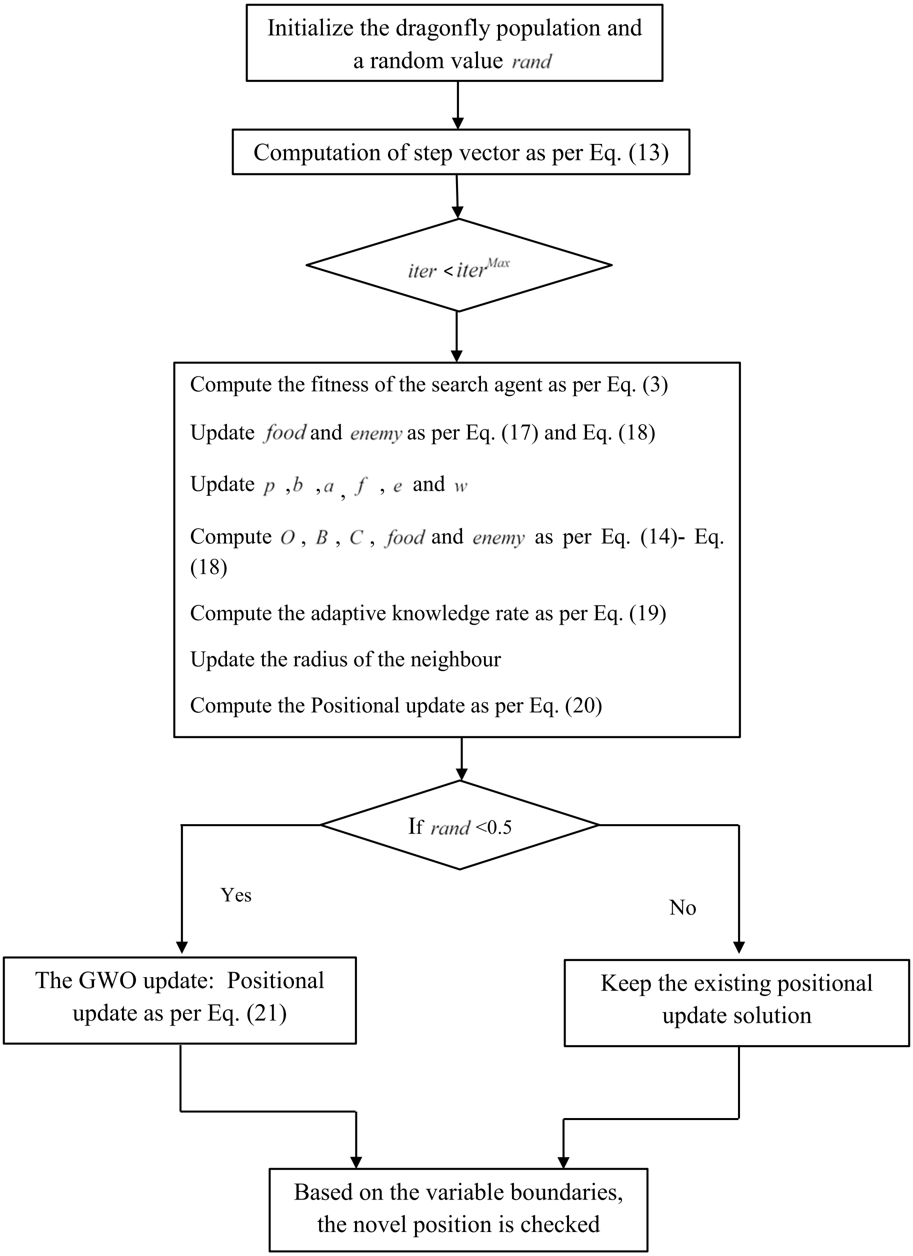


Fig. 4 Flow chart of the suggested DAGWO model

counts of VM in each PM). The DAGWO is evaluated over the existing works like PSO [29], WOA [45], DA [16], GWO [47],MGWO [30] andGA [18].

5.2 Convergence analysis

The convergence of the DAGWO&existing scheme is determined for scenario1 and scenario 2 throughchanging the iterations number from 20, 40, 60, 80, and 100, respectively. The results acquired with scenario1 and scenario 2 are shown in Fig. 5. As per the defined 4-fold objectives in Eq. (3), the schemewhich attains the least cost function is said to be the most appropriate method. At the least count of iteration, the cost function achievedvia the DAGWO &extant work is greater. When the number of iterations increases, the cost function of DAGWO, as well as existing work, gets minimized. In the case of scenario1, the cost function of DAGWO had attained a steep fall in between the range 0 to 20 count of iterations. In addition, when analyzing Fig.4 b, the cost function of the suggested&extant technique seem to be greater at the 3rd iteration. Till the 65th iterations, the cost function of the proposed seems was little worse than the existing one, Beyond the 65th iteration, the DAGWO had attained the least value as (~)0.125. Thus, The DAGWO clearly showed the small cost function that suggests that the proposed model may schedule the tasks more effectively.

5.3 Evaluation on energy consumption

Energy management is becoming more crucial in cloud storage as a result of rising energy prices and greater consumption of cloud computing resources. The call of energy-efficient solutions is the reduced the total energy usage of computing, storage, and communication devices. In data centres, optimum energy consumption is increasingly necessary. The current research tends on energy-efficient resource allocation. The energy required for executing a task must be lower, which leads to the expansion of network lifetime. The consumed energy by VM is represented in Fig. 6. In Fig. 6 a, the DAGWO seems to exhibit the least energy value while evaluating the tasks. For the task count =200, the DAGWO is 16.6%, 96.8%, 3.8%, 3.25%, 2.85%and 1.08%superior to the extant PSO, WOA, DA, GWO,MGWO & GA schemesfor optimal scheduling. On the other hand, the energy consumed by the DAGWO in scenario2 is fluctuating while the task execution. However, the overall performance shows that the proposed algorithm is more effective in solving the task scheduling issue with minimal energy consumption.

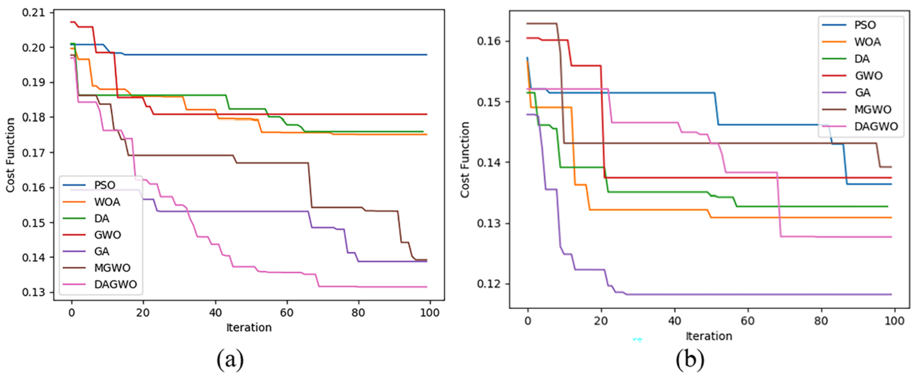


Fig. 5 Convergence analysis: DAGWO &extantschemes for (**a**) scenario 1 (VM = 30) & (**b**) scenario 2 (VM = 60)

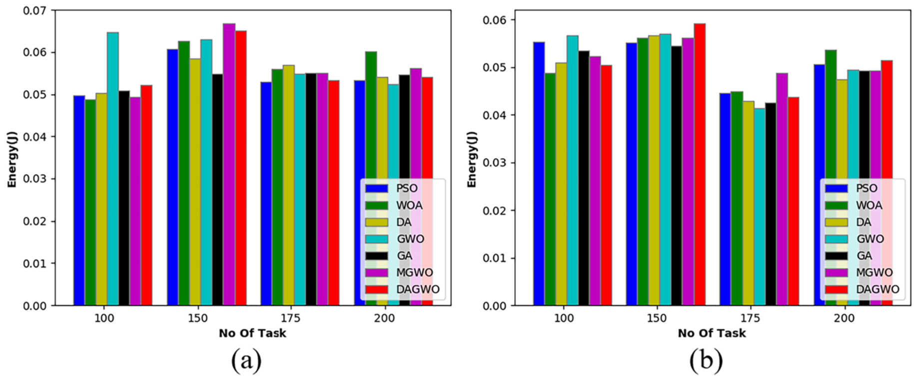


Fig. 6 Energy consumption evaluation: DAGWO &extantmodels for (**a**) scenario1 (VM = 30) & (**b**) scenario2 (VM = 60)

5.4 Evaluation on execution time

The computation time recorded via the suggested as well as an existing model is represented in Fig. 7. Further analyzing Fig. 7 a for task count = 200, the DAGWO is 11.1%, 6.9%, 12.2%, 42.85%, 23.07%, and 0.40%better than the extant PSO, WOA, DA, GWO, MGWO,&GA works with minimum execution time. Furthermore, on analyzingscenario2 at task count = 100, the DAGWO model is attained the least execution time (~ 0.28) than the extant PSO, WOA, DA, GWO,& MGWO models. Even though the execution time of the DAGWO seems to have fluctuated for certain task counts, the overall performance of the suggestedscheme is promising as well as it reveals the adopted scheme betterment.

5.5 Evaluation on makespan

Figure 8 depicts themakespan evaluation of the suggestedas well asextantschmes. On analyzing the outcomes, at task = 150, the DAGWO scheme is 25%, 52%, 33.3%, 45.45%,

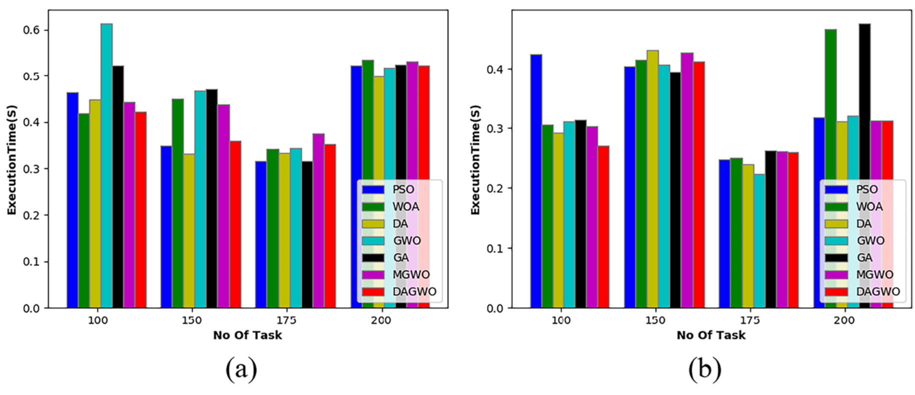


Fig. 7 Execution time evaluation: DAGWO &extantmodels for (**a**) scenario 1 (VM = 30) & (**b**) scenario2 (VM = 60)

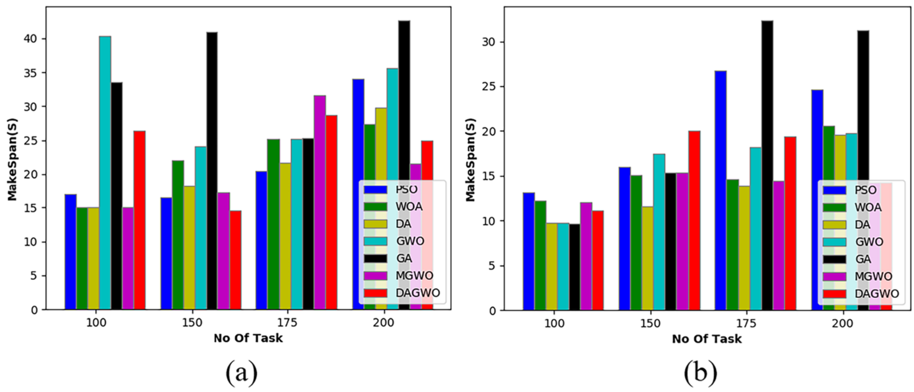


Fig. 8 Makespan evaluation: DAGWO &extant schemes for (**a**) scenario1 (VM = 30) & (**b**) scenario2 (VM = 60)

20% and 64.51% superior to the extant PSO, WOA, DA, GWO, MGWO,and GAapproaches respectively with minimal makespan. Then, forscenario2, the suggested modelis attained the small value at scheduling 200 numbers of tasks and it is smaller thanthe extantapproaches.

5.6 Evaluation on resource utilization

Minimal resource utilization is required during the scheduling & task execution in order to meet the given aim in Eq. (3). The outcomes of the suggestedas well as the extant approachesis depicted in Fig. 9. The DAGWO has scheduled and completed 175 counts of jobs with the least amount of resource use, according to the scenario 1 analysis. In this instance, the given work outperforms existing methodologies including PSO, WOA, DA, GWO, MGWO, and GA by 50%, 20%, 33.3%, 3.86%, and 3.21%, respectively. The given work, has recorded the lowest value of 0.05 in the scenario 2 condition, which is determined to be a better outcome when compared to the conventional schemes. The analysis so

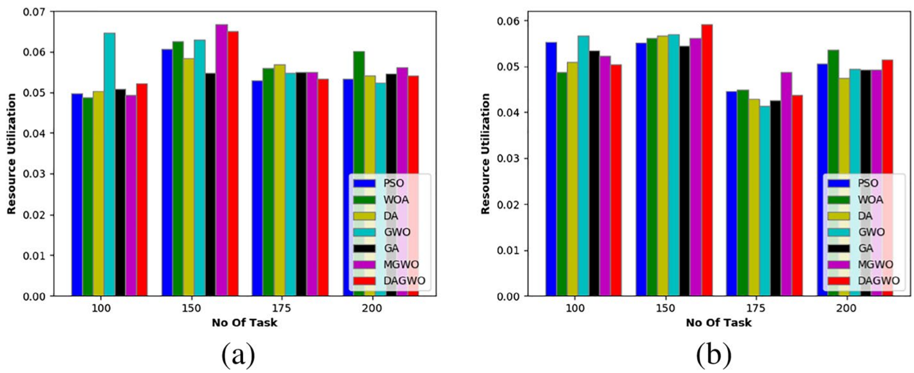


Fig. 9 Resource utilization evaluation: DAGWO &extant models for (**a**) scenario1 (VM = 30) & (**b**) scenario2 (VM = 60)

Table 2 Risk probabilityevaluation: DAGWO &extant schemesfor scenario1

Count of Tasks	PSO[10]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	38.264	33.635	33.746	21.447	24.610	55.849	16.839
150	21.996	19.613	18.366	21.577	19.800	31.633	23.174
175	35.997	37.352	31.179	39.325	24.666	38.613	27.438
200	52.141	55.639	48.461	44.231	55.119	54.519	45.906

demonstrated that the suggested model was superior for task scheduling while utilizing the fewest resources.

5.7 Evaluation on security or risk probability

The security analysis for adopted as well as extant approaches for scenarios is represented in Table 2 and 3. The findings demonstrate that the DAGWO balances excellent security with reduced risk. On analyzing scenario 1, the DAGWO attask = 100 is 16.83, and the extant model values are PSO = 38.2, WOA = 33.63, DA = 33.7, GWO = 21.4, MGWO = 24.6, and GA = 55.84. On observing all other tasks too, the risk probability of the suggestedshows to be lower than the conventional models. Further, in the case of scenario2, the risk probability of the DAGWO is 63.046 at task = 200 and this is the less value. Altogether, the analysis proves that the suggested model has the ability to process the scheduling as well as execution of tasks even in a secured manner.

5.8 Evaluation on throughput

The throughput is the greatest rate at which tasks may be completed in a given amount of time. It assesses the effectiveness of the scheduling method. Low response time and large execution rate result from a high throughput rate. Table 4 and 5 represents the throughput of the adoptedas well asextant schems at various tasks. In both scenarios, the throughput value of the suggested approach is increased with increasing the number of tasks. In Table 4, the throughput of the proposed work at scheduling tasks 200 is 36.11%, 33.03%, 29.26%, 44.34%, 0.34%, and 36.11% better than the existing PSO, WOA, DA, GWO, GA and MGWO approaches respectively. For scenario 2, the throughput of the adopted research at scheduling tasks 200 is2.72%, 11.21%, 7.44%, 12.96%, 10.57%, and 13.61% superior to the extantPSO, WOA, DA, GWO, GA and MGWO approaches respectively. Thus, the proposed model guarantees good throughput at various tasks.

Table 3 Risk probability evaluation: DAGWO &extant schemesfor scenario2

Count of Tasks	PSO[10]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	24.517	23.993	25.594	26.481	26.027	36.182	23.692
150	30.597	31.897	25.924	28.701	22.909	39.735	21.567
175	39.534	50.870	37.291	49.644	48.394	66.901	43.298
200	43.198	51.077	48.291	61.009	57.095	86.832	63.047

Table 4 Throughput (pps) evaluation: DAGWO &extant schemes for scenario1

No of Tasks	PSO[10]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	623.283	562.671	603.216	696.414	595.819	701.865	557.279
150	663.166	684.119	631.255	711.986	665.493	714.844	621.717
175	999.058	1080.647	1053.837	1082.442	1182.263	995.408	995.576
200	1073.911	1095.957	1021.057	965.206	1091.058	1076.208	956.885

5.9 Time complexity

The time consumed through the adopted as well as traditional schemes is depicted in Table 6 and 7 for scenario 1as well as scenario2. For scenario 1, the adopted DAGWO-for task 175 is 15.492%, 19.659%, 11.674%, 38.054%,41.12% and 9.186% superior to the extant PSO, WOA, DA, GWO, GA & MGWO approaches. In the case of Table 7, the computation time obtained by the developed DAGWO while scheduling tasks 200 is 0.195 s and it is 36.11%, 33.03%, 29.26%, 44.43%, 0.34%, and 36.11% superior to the extant PSO, WOA, DA, GWO, GA & MGWO methods. Hence, the superiority of the adopted scheme is verified with respect to computation time.

5.10 Rank analysis

The rank analysis of the adopted as well as existing schemes is shown in Table 8. This table shows the rank of the 4-objectives like "energy consumption, makespan, resource utilization cost and execution time" and also the overall rank of the methods are represented. On analyzing the results, the proposed model does not attain the first rank. But considering the overall analysis, the rank of the methods like PSO, WOA, DA, GWO, GA, MGWO, and proposed DAGWO are 7, 3, 2, 6, 5, 4 and 1. Therefore, the effectiveness of the proposed approach is proved successfully.

5.11 Discussions

The suggestedscheme considers the objectives along with the security constraints. The resultants show the efficiency of the proposed work and it is well suited for a multi-cloud environment. Our method attains high makespan, good resource utilization, less risk probability, low energy consumption, and computation time. Furthermore, the proposed model achieves high security than other existing methods with low risk. Sometimes the proposed model lacks its

Table 5 Throughput (pps) evaluation: DAGWO &extant schemesfor scenario2

No of Tasks	PSO[10]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	544.738	596.841	572.542	608.823	592.565	613.443	529.929
150	839.786	861.599	895.672	845.055	888.883	819.852	828.403
175	876.909	884.807	847.006	790.269	923.281	926.231	790.921
200	998.548	1058.586	974.607	1006.813	978.763	1079.772	979.557

Table 6 Computation time (s) evaluation: DAGWO &extant schemes for scenario1

Number of task	PSO [29]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	0.114329	0.107438	0.109697	0.126513	0.073097	0.164329	0.071259
150	0.126643	0.135024	0.125993	0.151539	0.092428	0.204329	0.106104
175	0.138179	0.145345	0.132205	0.188507	0.128584	0.198329	0.116771
200	0.220585	0.238271	0.203186	0.231051	0.183171	0.207329	0.176934

Table 7 Computation time (s)evaluation: DAGWO &extant schemesfor scenario2

Number of task	PSO [29]	WOA [45]	DA [16]	GWO [47]	MGWO [30]	GA [18]	DAGWO
100	0.097172	0.100926	0.147046	0.165688	0.071388	0.097172	0.111682
150	0.216606	0.251551	0.266266	0.273753	0.152369	0.216606	0.163678
175	0.238304	0.324933	0.25915	0.318829	0.201988	0.238304	0.200971
200	0.30624	0.292166	0.276618	0.351549	0.196334	0.306240	0.195666

Table 8 Rank analysis of the DAGWO and extant approaches

Method	Energy consumption	Makespan	Resource utilization cost	Execution time	Overall rank
PSO [29]	6	7	6	7	7
WOA [45]	1	6	1	4	3
DA [16]	3	2	3	2	2
GWO [47]	7	3	7	5	6
GA [18]	5	1	5	6	5
MGWO [30]	4	5	4	3	4
DAGWO	2	4	2	1	1

performance when compared to MGWO and this model is restricted by dynamic scheduling. In the future, we enhanced the suggested work to achieve improved results. Also, we consider some task features like deadlines and the data transmission between the workflow tasks.

6 Conclusion

Data transfer across application tasks is unavoidable in a virtualized cloud environment. If a virtual machine's security mechanism isn't strong enough, malicious activity can disrupt other tasks via altering intermediate data. This research introducesa secure task scheduling scheme for a multi-cloud environment. The study that was given dealt with using the optimization concept to allocate the tasks in the best possible way. As a result, the objectives are taken into account by the suggested optimal task scheduling frameworklike"makespan, execution time, and utilization cost along with security constraints". A novel risk probability evaluation was taken into account for ensuring security. For optimal scheduling, a uniqueapproach referred to as DAGWO that hybridizes the concepts of GWO & DA was introduced. The DAGWO is determined over the existing works in terms of convergence, energy, makespan as well. The experimental findings reveal that, when compared to existing methods, our method effectively reduces the security risk while keeping a respectable completion time. However, the execution time, makespan, utilization cost of the DAGWO model seems to oscillate in terms of performance, the overall objective is found to be achieved.

Funding None

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this study

Declarations

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

Informed consent not applicable

Ethical approval This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Abdulhamid SI, Abd Latiff MS, Abdul-Salaam G, Hussain Madni SH (2016) Secure Scientific Applications Scheduling Technique for Cloud Computing Environment Using Global League Championship Algorithm",Department of Cyber Security Science
2. Abdullahi M, Ngadi MA, Abdulhamid SM (2016) Symbiotic Organism Search optimization based task scheduling in cloud computing environment. Futur Gener Comput Syst 56:640–650
3. Alexandridis G, Tagaris T, Siolas G, Stafylopatis A (2019) From free-text user reviews to product recommendation using paragraph vectors and matrix factorization. In: Companion Proceedings of The 2019 World Wide Web Conference, pp. 335–343
4. Ashok Kumar C, Vimala R (2020) Load Balancing in Cloud Environment Exploiting Hybridization of Chicken Swarm and Enhanced Raven Roosting Optimization Algorithm. Multimed Res 3(1):45–55
5. Brammya, Deepa TA (2019) Job Sceduling in Cloud Environment using Lion Algorithm. J Networ Commun Syst 2(1):1–14

6. Chen H, Zhu X, Qiu D, Liu L, Du Z (2017) Scheduling for Workflows with Security-Sensitive Intermediate Data by Selective Tasks Duplication in Clouds. IEEE Trans Para Distrib Syst 28(9):2674–2688. https://doi.org/10.1109/TPDS.2017.2678507
7. M. Chen, B. Liang and M. Dong, "Multi-User Multi-Task Offloading and Resource Allocation in Mobile Cloud Systems," IEEE Trans Wirel Commun, vol. 17, no. 10, pp. 6790–6805, Oct. 2018.
8. 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. https://doi.org/10.1109/ACCESS.2020.3032545
9. Farid M, Latip R, Hussin M, Abdul Hamid NAW (2020) Scheduling Scientific Workflow Using Multi-Objective Algorithm With Fuzzy Resource Utilization in Multi-Cloud Environment. IEEE Access 8:24309–24322. https://doi.org/10.1109/ACCESS.2020.2970475
10. Ge J, He Q, Fang Y. "Cloud computing task scheduling strategy based on improved differential evolution algorithm." In: AIP Conference Proceedings, vol. 1834, no. 1, p. 040038. AIP Publishing LLC, 2017.
11. George Amalarathinam DI, Madhu Priya J (2018) Survey on Data Security in Multi-Cloud Environment. Int J Pure App Math 118(6):323–334
12. Gopal A (2020) Hybrid classifier: Brain Tumor Classification and Segmentation using Genetic-based Grey Wolf optimization. Multimed Res 3(2)
13. Grzonka D, Kołodziej J, Tao J, Khan SU (2015) Artificial Neural Network support to monitoring of the evolutionary driven security aware scheduling in computational distributed environments. Futur Gener Comput Syst 51:72–86
14. Ismayilov G (2020) Haluk Rahmi Topcuoglu, "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing". Futur Gener Comput Syst 102:307–322
15. Jadhav AN, Gomathi N (2019) DIGWO: Hybridization of Dragonfly Algorithm with Improved Grey Wolf Optimization Algorithm for Data Clustering. Multimed Res 2(3):1–11
16. Jafari M, Chaleshtari MHB (2017) Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out. Eur J Mechanics A/Solids 66:1–14
17. Fredy Juarez, Jorge Ejarque, Rosa M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing", Futur Gener Comput Syst, vol. 78, Part 1, pp. 257–271, 2018.
18. Keshanchi B, Souri A, Navimipour NJ (2017) An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing. J Syst Softw 124:1–21
19. Kokilavani S, Vignesh MD, Surya E, Narendran A, Prasanalakshmi B (2014) Enhanced biometric smart key scheme for smart card authentication. In: In 2014 International Conference on Communication and Signal Processing. IEEE, pp 1589–1592
20. Lavanya M, Shanthi B, Saravanan S (2020) Multi objective task scheduling algorithm based on SLA and processing time suitable for cloud environment. Computer Communications, vol. 151, pp. 183–195
21. Lee J-W, Jang G, Jung H, Lee J-G, Lee U (2019) Maximizing MapReduce job speed and reliability in the mobile cloud by optimizing task allocation. Pervasive and Mobile Computing, vol. 60, Article 101082
22. Liu Y, Xun X, Zhang L, Wang L, Zhong RY (2017) Workload-based multi-task scheduling in cloud manufacturing. Robot Comput Integr Manuf 45:3–20
23. Mansouri N, Zade BMH, Javidi MM (2019) Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. Comput Ind Eng 130:597–633
24. Mapetu JP, Buanga ZC, Kong L (2019) Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. Appl Intell 49(9):3308–3330
25. Marsaline Beno M, Valarmathi IR, Swamy SM, Rajakumar BR (2014) Threshold prediction for segmenting tumour from brain MRI scans. Int J Imaging Syst Technol 24(2):129–137. https://doi.org/10.1002/ima.22087
26. Masdari M, Khoshnevis A (2020) A survey and classification of the workload forecasting methods in cloud computing. Clust Comput 23(4):2399–2424
27. Masdari M, Zangakani M (2020) Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities. J Supercomput 76(1):499–535
28. Masdari M, ValiKardan S, Shahi Z, Azar SI (2016) Towards workflow scheduling in cloud computing: a comprehensive analysis. J Netw Comput Appl 66:64–82
29. Mishra SK et al (2020) Energy-Aware Task Allocation for Multi-Cloud Networks. IEEE Access 8:178825–178834. https://doi.org/10.1109/ACCESS.2020.302687
30. Mulge MY (2019) Optimization of Task Scheduling Algorithm Using Modified Mean Grey-Wolf. Int J Int Engin Syst 12(4):192–200
31. Neelima P, Reddy ARM (2020) An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. Clust Comput. https://doi.org/10.1007/s10586-020-03054-w
32. Neelima P, Reddy A (2020) An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. Clust Comput 23(4):2891–2899

33. Ninu Preetha NS, Brammya G, Ramya R, Praveena S, Binu D, Rajakumar BR (2018) Grey Wolf Optimisation-based Feature Selection and Classification for Facial Emotion Recognition. IET Biomet 7(5):490–499. https://doi.org/10.1049/iet-bmt.2017.0160
34. Niu S, Zhai J, Ma X, Tang X, Chen W, Zheng W (2016) Building Semi-Elastic Virtual Clusters for Cost-Effective HPC Cloud Resource Provisioning. IEEE Trans Para Distrib Syst 27(7):1915–1928. https://doi.org/10.1109/TPDS.2015.2476459
35. Panda SK, Gupta I,Jana PK (2017) "Task scheduling algorithms for multi-cloud systems: allocation-aware approach", Inf Syst Front
36. Panda SK, Gupta I, Jana PK (2019) Task scheduling algorithms for multi-cloud systems: allocation-aware approach. Inf Syst Front 21:241–259
37. Pang S, Li W, He H, Shan Z, Wang X (2019) An EDA-GA Hybrid Algorithm for Multi-Objective Task Scheduling in Cloud Computing. IEEE Access 7:146379–146389
38. Rahman CM, Rashid TA (2019) Dragonfly Algorithm and Its Applications in Applied Science Survey",Computational Intelligence and Neuroscience, vol. 2019, Article ID 9293617, 21 pages
39. Rajakumar BR (2013) Impact of Static and Adaptive Mutation Techniques on Genetic Algorithm. Int J Hybrid Int Syst 10(1):11–22. https://doi.org/10.3233/HIS-120161
40. Rajakumar BR (2018) Optimization using lion algorithm: a biological inspiration from lion's social behavior. Evol Int, Special Issue Nat Inspired Algo High Perform Comput Comput Vis 11(1–2):31–52. https://doi.org/10.1007/s12065-018-0168-y
41. Ren Z, Taghipour A, Canel-Depitre B (2016) Information sharing in supply chain under uncertainty. In: 2016 6th International Conference on Information Communication and Management (ICICM). IEEE, pp 67–71
42. Rjoub G, Bentahar J (2020) Omar Abdel Wahab, "Big Trust Scheduling: Trust-aware big data task scheduling approach in cloud computing environments". Futur Gener Comput Syst 110:1079–1097
43. Roy RG (2019) Escheduling based congestion management method using hybrid Grey Wolf optimization-grasshopper optimization algorithm in power system. J Comput Mech, Power Syst Cont 2(1):9–18
44. Salman T (2015) On Securing Multi-Clouds: Survey on Advances and Current Challenges
45. Sanaj MS, Joe Prathap PM (2019) Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. Engineering Science and Technology, an International JournalIn press, corrected proof Available online 22
46. Senthilnathan R, Nithya M (2018) A Trust model and quality of service based heuristic scheduling in cloud using genetic algorithm. Int J Pure App Math 119(16):1007–1018
47. Seyedali Mirjalili A, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61
48. Shishido HY, Estrella JC, Toledo CFM, Arantes MS (2018) Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. Comput Electr Eng 69:378–394
49. Simic V, Stojanovic B, Ivanovic M, "Optimizing the performance of optimization in the cloud environment–An intelligent auto-scaling approach", Futur Gener Comput Syst, vol. 101, pp. 909–920, December 2019.
50. Subramanyam TC, Ram SST, Subrahmanyam JBV (2018) HGAGWO: A Multi-objective optimal positioning and sizing of fuel cells in DG systems. J Comput Mechan, Power Syst Contr 1(1):34–44
51. Uma Maheswari S (2016) Security and Privacy Enhancing Multicloud Architectures. Int J Engin Sci Comput 6(5)
52. Vinolin V (2019) Breast Cancer Detection by Optimal Classification using GWO Algorithm. Multimed Res 2(2):10–18
53. Yi Zhang Y, Liu JZ, Sun J, Li K (2020) Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing. Futur Gener Comput Syst 112:148–161
54. Zeng L, Veeravalli B, Li X (2015) SABA: A security-aware and budget-aware workflow scheduling strategy in clouds. J of Para Distribut Comput 75:141–151
55. Zhou C, Li X, Yang S, Tian Y (2020) Risk-Based Scheduling of Security Tasks in Industrial Control Systems With Consideration of Safety. IEEE Trans Industr Inform 16(5):3112–3123. https://doi.org/10.1109/TII.2019.2903224