



An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment

Jabir Kakkottakath Valappil Thekkepurayil¹ · David Peter Suseelan¹ · Preetha Mathew Keerikkattil¹

Received: 22 October 2020 / Revised: 1 February 2021 / Accepted: 14 March 2021 / Published online: 31 March 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Cloud computing is an emerging distributed computing model that offers computational capability over internet. Cloud provides a huge level collection of powerful and scalable computational resources for computation and data-intensive large scale workflow deployment. For business as well as scientific applications, optimal scheduling of workflow is emerged as a major concern. Optimization of scheduling process leads to the reduction of execution time, cost, etc. So, this paper presents an enhanced recent ant-lion optimization (ALO) algorithm hybridized with popular particle swarm optimization (PSO) algorithm to optimize a workflow scheduling specifically for cloud. A security approach called Data Encryption Standard (DES) is used for encoding the cloud information while scheduling is carried out. The research aims to contribute an enhanced workflow scheduling more safely than the existing frameworks. Enhancement procedures are evaluated in terms of cost, load, and makespan. The simulation procedures are done by utilizing the CloudSim tool. The proposed hybrid optimization results contrasted with well-known existing approaches. The existing round-robin (RR), ALO and PSO methods are selected to compare and identify the potency of the proposed system. The outcomes indicated that the proposed technique minimizes the cost by 9.8% of GA-PSO, 10% of PSO, 20% of ALO, 30% of RR and 12% of GA. Load balancing and makespan of the proposed method reduces by 8% than GA-PSO, 10% than ALO, 20% than PSO, 35% than RR and 45% than GA. The energy consumption and reliability performance are also reasonably well.

Keywords Workflow scheduling · Cloud computing · Hybridization · Multi-objective optimization · ALO · PSO · Data encryption standard

Abbreviations

IaaS	Infrastructure as a Service
VM	Virtual machine
CSP	Cloud Service Provider
QoS	Quality of Service
GSA	Gravitational Search Algorithm
HEFT	Heterogeneous Earliest Finish Time
PEFT	Predict Earliest Finish Time algorithm
MOGA	Multi-objective Genetic Algorithm
MOPSO	Multi-objective Particle Swarm Optimization Algorithm
CPM	Cost Prediction Matrix
SCPS	Secured Cost Prediction based scheduling
DAG	Direct Acyclic Graph
MOP	Multi-objective Optimization Problem

DES	Data Encryption Standard
PSO	Particle Swarm Optimization
ALO	Ant-lion Optimization

1 Introduction

Cloud computing is a highly internet-based and versatile computing environment which gives on-demand services and working platforms to fulfill the computational needs of different users [1]. Infrastructure as a Service (IaaS) is a standout amongst a widely recognized cloud service framework which provides proficient and adaptable computational resources to the users. These models discharge cloud infrastructure as virtual machines (VMs) [2]. Clients being able to access a practically infinite amount of resources with less possession cost for application execution. Huge scale workloads can be kept running on the VMs facilitated by the cloud infrastructure [3]. Users may

✉ Jabir Kakkottakath Valappil Thekkepurayil
kvtjabir@cusat.ac.in

¹ Cochin University of Science and Technology, Kochi, Kerala, India

use services with the rate fixed by their CSP (Cloud Service Provider) for specific requirements. These services could be delivered at any time [4].

Workflow logically portrays data-intensive applications that are hosted and facilitated on the cloud infrastructure. The tasks and information conditions among the works are depicted in the workflow models [5]. Normally, the workflow arranges the procedure in a Directed Acyclic Graph (DAG) structure, wherein every single node denotes the constituent task, and edges mean the inter-task conditions [6]. Workflows are composed of numerous interdependent data-intensive computational tasks, and they require high-performance computing resources for efficient execution. Hence the scheduling and mapping of these tasks to suitable resources in a proper way are significant in dealing with the workflow executions [7]. The greater part of the scheduling algorithms used in cloud model considers disregard security and resource execution issues of workflow implementation in the cloud [8].

The cloud security system comprises a lot of arrangements, controls, methods, and advancements, which are cooperated to ensure secure information in cloud-based frameworks and infrastructure [9]. Typically, cloud security procedures are assumed to be a mutual responsibility among solution provider and the owner. Cloud security methods are intended to support consistency, protect data, and confirm the safety of client privacy. Leakage of data, alterations of sensitive information by malicious attacks are considered as the two major threats in the cloud computing environment [10]. Due to shared infrastructure, security concerns while scheduling workflows is considered as one of the main complexity in the cloud [11]. Recently, more considerations are paid in the safety efforts of the cloud. In data center, VMs running on a physical machines (PMs) are considered as an essential element in cloud. Nowadays, diverse security components are available in VMs to assure the controls. Execution of security procedures in cloud computing reduces security risk and threats [12].

The objective function can be limited or maximized to fulfill clients' QoS requirements by the allocation of tasks to appropriate resources in cloud computing. The QoS requirement based workflow scheduling currently incorporates the time, cost, security, load, and success rate, etc. [13]. But many of the researches in the field confined to single or bi objective optimization of QoS parameters. This substantiates the study on multi objective optimization as more QoS objectives are to be optimized. The major QoS objectives in normal optimizations are time and cost. Load balancing is a serious issue which leads to performance degradation in system [14]. Load balancing recognized as parameter for multi objective optimization in this work. User requirement also has a significant role in deciding the QoS objectives [15]. Distinctive enhancement algorithms

like a PSO, simulated annealing (SA), genetic algorithm (GA), etc. are used in a workflow applications for the effective optimization of QoS objectives. These algorithms bring enhanced results; however, some issues may arise like the overhead due to time-consuming procedures [16]. Many research used only heuristic methods for workflow scheduling. The scope of meta-heuristic methods for workflow optimization is still not completely explored and extensive researches are continuing in the field. Each meta-heuristic algorithm has its own advantages and disadvantages therefore hybrid method is emerging as a solution. The vast majority of the workflow scheduling examinations concentrates the expense and makespan. The security is one of the QoS constraints that is not properly addressed in past works [17]. In these contexts, the main contributions of our proposed methodology are explained as below:

- Meta-heuristic solution to execute workflows in cloud computing systems with multi objective optimization.
- A scheduling framework based on a hybrid ALO-PSO procedure is developed to perform an efficient workflow scheduling in the cloud computing model with data security.
- A low complexity scheduling algorithm based on meta heuristic hybrid technique with upstanding results in terms of load, cost and makespan.

Towards this, the efficient particle swarm algorithm is hybridized with recent ALO and outcomes are compared with state of the art algorithms to distinguish an impacts of proposed strategy. The security algorithms in this work shield the information or data from alteration and leakage. The remaining paper is systematized as follows: In Sect. 2, related work is described. Problem formulation and framework models are demonstrated in Sect. 3. Section 4 contains the simulation strategies and anticipated outcomes. Section 5 deals with test outcomes and evaluation. At last, Sect. 6 provides the conclusion and recommendations for future work.

2 Literature survey

Some of the previous researches related to the present study are discussed in the below sections.

Due to its wide business and scientific applications, cloud-based workflow scheduling had gained major attention. Different researchers were used various heuristic and meta-heuristic strategies for scheduling by considering a few issues, for example, protection of energy, cost, and makespan. Choudhary et al. [18] exhibited a calculation for the processing, which deals with the reduction of cost and makespan. To schedule workflow applications, hybrid of heterogeneous earliest finish time (HEFT) algorithm and

gravitational search algorithm (GSA) was used. Another issue named as equivalence in cost time made the objectives enhancement additionally sensible. Financial cost proportion, schedule lengths were considered for the performance measurements to contrast with the existing strategies. The ultimate results showed that the used strategy gives better results.

Manasrah and Hanan [19] developed a hybrid GA-PSO procedure for workflow scheduling in the cloud framework. This hybrid method effectively achieves the task allocation to the resources. Moreover, it considered some of the parameters like cost, makespan and load balance rate as an objective function to perform the task allocation in cloud environment. This hybrid algorithm was differentiated by the properties of GA as well as PSO algorithm. The evaluation of proposed methodology was tested with different sizes of workflow applications. The multi-objective task scheduling based on a hybrid EDA-GA algorithm was developed by Pang et al. [20]. EDA stands for Estimation of Distribution Algorithm. Initially, the sampling method and probability model of EDA were utilized to create the specified number of feasible solutions. Then, the search range of solutions was generated to initiate the crossover and mutation function. So, based on the hybrid algorithm functions, final task allocation to VM was achieved.

Workflow scheduling in the cloud is very challenging due to the versatility and heterogeneity of cloud resources. Optimization of execution cost and execution time are two prominent basic issues for scheduling resources in the cloud. Chen et al. [21] modeled a cloud workflow scheduling as a multi-objective (MO) enhancement issue with the scheduling objectives, minimization of execution time, and execution cost. The proposed method was an ant colony system (ACS) using multiple populations for multiple objectives. The scheduling was based on two colonies for those two objectives of time and cost.

Apart from the out-of-date objectives like budget, deadline, and makespan, in this proposed multi-objective GA (MOGA), Rehman et al. [22] considered the energy consumption aspect also. To provide energy-efficient solutions while optimization, dynamic voltage frequency scaling method was utilized in the study. A gap search calculation that finds the gap between continuous engagement periods of VMs and the optimal use of cloud resources. The results of MOGA were equated with multi-objective PSO (MOPSO) through similar objective constraints as that of MOGA. The MOGA gave better results when contrasted with other GA based algorithms which considered objectives like makespan, cost, and deadline individually.

Nowadays, meta-heuristic algorithms are emerged as a prime choice for workflow scheduling in the cloud because of its NP-hard nature. Shishido et al. [23] examined the

efficiency of meta-heuristic methods to schedule workflows in the cloud. This study assessed the impacts of both PSO enhancement and GA for workflow scheduling optimization. A cost-aware workflow scheduling problem was adopted towards measuring the competency of meta-heuristic methodology and the experiments were conducted with PSO, GA, and Multi Population GA meta-heuristic. Meta-heuristic algorithms were evaluated on minimization of cost and response time objectives. These algorithms returned better schedules that decrease the expense inside a sensible timeframe.

Angela et al. [24] suggested secured cost prediction-based scheduling (SCPS) method, which emphasizes security while minimizing makespan in workflow scheduling. Even though cloud services could be successfully utilized to execute big, information and computation-exhaustive scientific workflow requests, security remains a major concern. Cost prediction matrix (CPM) for expense estimation and a fuzzy based model for choosing appropriate VMs based on security concerns are the basic elements of this presented methodology.

For scientific workflow scheduling, Sharma and Rashid [25] developed a hybrid PSO algorithm in the cloud computing model. This research hybridized PSO algorithm with Predict Earliest Finish Time (PEFT) technique to make better scheduling process. Here, initial population was generated by PSO algorithm. Performance of proposed methodology was estimated in terms of cost and makespan. In cloud computing, Deadline-aware and Cost-effective Hybrid Genetic Task Scheduling (DCHG-TS) method was developed by Iranmanesh and Naji [26] for scientific workflow scheduling. In this GA, they used new genetic operators and modified genetic operators to enhance the load balancing routine. At execution time, this research applies a load balancing method to exploit the resources.

For multi-objective task scheduling problem in cloud environment, Abualigah and Diabat [27] developed a new hybrid ant lion optimization (ALO) procedure. Elite based differential evolution method was hybridized with ALO to resolve multi-objective task scheduling difficulty in which the proposed method was named as MALO. The multi-objective problem was derived to maximize resource utilization as well as to minimize makespan. Moreover, exploitation ability of ALO algorithm was improved by the elite based differential evolution technique. Moth-flame optimization algorithm (MFO) [28] was developed to perform an efficient task scheduling (TS) method for cyber-physical system (CPS) applications in fog computing. This algorithm considered minimization of transfer time and task execution as fitness function in optimization algorithm.

Garg et al. [29] developed a reliable and energy efficient workflow scheduling in cloud model. Energy consumption

of a whole system was condensed by an efficient Dynamic voltage and frequency scaling (DVFS) method. During application execution, negative effect of DVFS increases the transient faults in system reliability. This paper developed a new scheduling algorithm to overcome the above-mentioned problems. There were four stages in the scheduling algorithm which were named as priority estimation, task clustering, target time distribution and conveying cluster to processing component that have proper frequency or voltage levels. An enhanced GA was developed by Keshanchi et al. [30] to perform a task scheduling in cloud environment. Accuracy of the proposed algorithm was checked by behavioral modeling method. Then, predictable stipulations of the proposed method were extracted in the form of linear temporal logic (LTL) formulas. Labeled Transition System (LTS) method also utilized to confirm the performance of suggested technique. Comparative analysis of existing methods is listed in Table 1.

Applications of workflow have variety of different tasks and composite structure. Every single task is processed by entering information, accessing software, processing or storage functions. It is difficult to achieve a good trade off solution among cost execution time. These are considered as a reasons for a workflow scheduling to be a NP-hard problem. For the selection of proper resources, it is very essential to develop an efficient algorithms for workflow execution. Past investigations take a lot of efforts in workflow scheduling to schedule the works properly. From the literature review, it is evident that almost all experimentations were done with single objective or bi-objective scheduling only. No single investigation has incorporated with all the parameters into one algorithm, as this study proposes.

Moreover, they have some limitations like time, maximum cost, QoS constraints and computational complexity. The security practices in workflow scheduling are

Table 1 Comparative analysis of existing methods

Author and year	Technique	Workload	Evaluation tool	Performance metric
Choudhary et al. [18], 2018	Hybrid GSA with HEFT	Real world scientific workflow applications (Montage, Cybershake, etc.)	C + + coding environment	Monetary cost ratio and schedule length ratio
Manasrah and Hanan [19], 2018	Hybrid GA-PSO	Real world scientific workflow applications	CloudSim	Makespan, execution cost and load balance rate
Pang et al. [20], 2019	Hybrid EDA-GA algorithm	Random workload	CloudSim	Load balancing degree (DBL), completion time
Chen et al. [21], 2019	MOACS technique	Real world scientific workflow applications	Amazon EC2 cloud platform	Run time
Rehman et al. [22], 2018	MOGA algorithm	HC-test bench	Java	Makespan, cost and energy
Shishido et al. [23], 2018	PSO-GA	Real world scientific workflow applications	CloudSim	Cost estimation
Angela et al. [24], 2018	SCPS method	Real world scientific workflow applications (Montage, Cybershake, etc..)	WorkflowSim	Makespan, Scheduling length ratio, communication to computation ratio
Sharma and Rashid [25], 2020	PPSO algorithm	Real world scientific workflow applications	Netbeans workflow simulator	Execution time and cost parameter
Naji [26], 2020	DCHG-TS	Both real-world workflows and synthesized workflows	CloudSim	Cost and Deadline factor
Abualigah and Diabat [27], 2020	MALO technique	Random workload	CloudSim	Degree of imbalance and Makespan
Ghobaei-Arani et al. [28], 2020	TS-MFO algorithm	Random workload	iFogSim toolkit and CloudSim	Execution time, transfer time and makespan
Garg et al. [29], 2019	DVFS	Random workload	CloudSim	Energy consumption and reliability
Keshanchi et al. [30], 2017	N-GA method	Random workload	C# language on Azure cloud environment	Makespan, execution time, efficiency and schedule length ratio (SLR)

mentioned very rarely in past researches. But, the proposed methodology utilizes DES algorithm for security in workflow applications. Moreover, the present investigation shows the impact of a hybrid meta-heuristic algorithm in multi-objective optimization of workflows. The optimization of cost, makespan, and load are the objectives, and it additionally provides security to the data during scheduling. A few algorithms are weak in local search, and others are weak in global search optimization. Hence hybridization of diverse meta-heuristic method seems to provide better results for workflow scheduling problems. In hybridization, the first optimization procedure is applied to elect the ideal weight parameter. In that algorithm, the presence of random parameter may increase the number of iterations. So, in order to reduce this, we hybrid another optimization algorithm that shows less computation time. Implementation of a hybrid system will improve a deficiencies of previous approaches. ALO is a recent meta-heuristic method which exhibits high exploration and convergence rate [36]. The ALO and PSO are claimed to be dominant than GA. Then the hybridization of ALO and PSO is expected to provide better optimization results.

3 System model

This section describes the cloud resource and workflow model utilized in this experimentation.

3.1 Model of workflow

DAG is commonly used to represent the structure of workflow. It is demonstrated as $W = (T, D)$ in which $T = \{T_0, T_1, T_2, \dots, T_n\}$ signifies the collection of n tasks in a workflow. Among the tasks, $D = \{(T_i, T_j) | T_i, T_j \in T\}$ demonstrates the set of data flow conditions. These are represented by (T_i, T_j) , which shows the constraint called precedence in between T_i and T_j . Tasks may have various predecessors and successors, the immediate predecessors and successors task set of T_i is specified by $\text{Pred}(T_i)$ and $\text{Succ}(T_i)$ respectively.

$$\text{Pred}(T_j) = \{T_i | (T_i, T_j) \in D\} \quad (1)$$

$$\text{Succ}(T_i) = \{T_j | (T_i, T_j) \in D\} \quad (2)$$

The task with no predecessor is termed as entry task, T_{entry} which is indicated by,

$$\text{Pred}(T_{\text{entry}}) = \varnothing \quad (3)$$

While tasks with no successors is named as exit task T_{exit} and shown by,

$$\text{Succ}(T_{\text{exit}}) = \varnothing \quad (4)$$

Generally, all workflow scheduling needs a DAG with single T_{exit} and T_{entry} . These can be accomplished effectively by including pseudo T_{exit} and T_{entry} having zero weight to DAG. Same assumptions are followed in every workflow and have only one T_{entry} and T_{exit} .

3.2 Cloud resource model

The computational resources in an IaaS platform are offered by VMs. Instances are the running VMs and different type of instances are being provided by IaaS. These instance types are of varying computing capacity, memory, and bandwidth combinations. For different instance types, the capacity of the CPU will decide the execution time of a task, and a bandwidth influences data communication time between tasks. User can use an infinite number of instances and it is indicated by an infinite set $Is = \{Is_0, Is_1, \dots, Is_a\}$. In an IaaS platform, the present instance types are indicated by a set of types $Ps = \{Ps_0, Ps_1, \dots, Ps_n\}$ where n is the number of types. An instance can run only one task at a time.

Compute unit is utilized to mention CPU capabilities of different instance types. Let $\text{Compute}(Ps_i)$ be the compute unit of the instance type Ps_i . The actual running time of a task in multi-core CPUs are distinguished as,

$$AR_{\text{time}}(T_i) = \frac{\text{refer time}(T_i)}{\text{Compute}(Ps_j)} \quad (5)$$

where $AR_{\text{time}}(T_i)$ is actual task running time of T_i . The reference time of execution for a task T_i is $\text{refer time}(T_i)$ in which it is mentioned as the time taken to execute the task on an instance whose compute unit equals to one [31]. The total bandwidth (BW) utilization is defined as the percentage of utilized bandwidth from the total available bandwidth. The given below expression is applied to calculate the bandwidth utilization of each VM.

$$BW_{\text{utilization}} = \frac{(\text{actual used BW in VM}_i)}{(\text{Total BW capacity of VM}_i)} \times 100 \quad (6)$$

Memory resource is assumed as one of the main resources in cloud computing. It is mainly utilized to satisfy the service request in the cloud. The capacity of memory is calculated based on the given below expression.

$$\text{Memory}_{\text{capacity}} = \frac{\text{utilized memory of VM}_i}{\text{Total memory of VM}_i} \quad (7)$$

3.3 Energy and system reliability model

Energy consumption in cloud data center is mainly from the usage of CPU, memory, network interfaces and disk storage. CPU consumes more power than any other

resources. Here, the energy consumption of single task t_i is represented in the given below expression.

$$E_i(f_{r,op}) = (P_{ind_i} + C_{eff}f_{r,op}^3) \frac{et_i}{f_{r,op}} = P_{ind_i} \frac{et_i}{f_{r,op}} + C_{eff}f_{r,op}^2 et_i \tag{8}$$

Here, implementation time of the task t_i at maximum operating frequency is mentioned as et_i , P_{ind_i} mentions the frequency independent power consumption, operating frequency level of processor is mentioned as $f_{r,op}$, and effective load capacitance is denoted as C_{eff} . Finally, the overall energy consumption is calculated based on the given below expression [29].

$$E_{total} = \sum_{i=1}^n E_i(f_{r,op}) \tag{9}$$

For computation intensive applications, reliability is considered as an important factor. Reliability of the system is defined as the probability of execution of task without any failure. So, this paper considered the poison distributed fault rate for reliability calculation. The reliability of task t_i with execution time et_i and operating frequency $f_{r,op}$ is mentioned in the given below expression

$$Rel_{t_i}(f_{r,op}) = e^{-\lambda(f_{r,op}) \cdot \frac{et_i}{f_{r,op}}} \tag{10}$$

The consistency of an application involving n tasks which is given as product of reliabilities of all its tasks and it is described in the given below expression [29].

$$Rel_G = \prod_{i=1}^n Rel_{t_i}(f_{r,op}) \tag{11}$$

3.4 Problem definition of workflow scheduling

An overview of multi-objective workflow scheduling problems is discussed in this section. These problems are considered to solve workflow scheduling in the proposed methodology. In this experimentation, the different metrics like makespan, load and cost are considered as main issues. The security efforts are trying to ensure data security while scheduling. Calculation of cost, makespan, and load are given initially for the problem definition.

Let $S_i(T_i)$ and $F_i(T_i)$ indicates the start and finish time of T_i . The starting time of a task is related to finishing time of all predecessors $Pred(T_i)$, communication time $Com_i(T_j, T_i)$ among its predecessors and itself, and completion time $F_i(T_j)$ of the earlier task which implemented on a same instance. Finish time $F_i(T_i)$ of task T_i are estimated as,

$$F_i(T_i) = S_i(T_i) + AR_{time}(T_i) = \max \left\{ Avail(Insta(T_i)), \max_{T_j \in Pred(T_i)} (F_i(T_j) + Com_i(T_j, T_i)) \right\} + AR_{time}(T_i) \tag{12}$$

The available time of instance, in which task T_i executes on can be given as $Avail(Insta(T_i))$ and this varies during scheduling. Beginning time of entry task T_{entry} be zero then finish time of T_{entry} be also zero. Completion time of exit task is distinguished as makespan and expressed as

$$Makespan = F_i(T_{exit}) \tag{13}$$

For evaluating cost, let $Cst = \{Cst_1, Cst_2, \dots, Cst_H\}$ be taken as the set of pricing schemes towards consuming services. $Monetary_{cst} = (Cst_h, Type, Run_{Insta})$ be the cost for running instance Run_{Insta} with instance type $Type$ and corresponding cost scheme Cst_H . The total monetary cost for executing all tasks in workflows is shown as,

$$Cost = \sum Monetary_{cst}(Cst_h, Type(Run_{Insta}), Run_{Insta}) \tag{14}$$

The next problem is load balancing. Load balancing can be characterized by the length of the task, the capacity of cloud resources, task dependency [41]. Proper load balancing can enhance the usage of resources and thereby forms effective schedules. The sum of all load in VM scan be demonstrated as,

$$Ld = \sum_{i=1}^k load_i \tag{15}$$

where i denotes a number of VMs in data center. Given below expression states the Load per unit capacity (L_{puc}).

$$L_{puc} = \frac{Ld}{\sum_{i=1}^m Cap_i} \tag{16}$$

$$Threshold\ value, \quad Tresh_i = L_{puc} * Cap_i \tag{17}$$

where capability of node is mentioned as Cap_i . Load imbalance factor of specific VM is shown as

$$VM_{capacity} \begin{cases} < |Tresh_i - \sum_{i=1}^k load_i| \text{Underloaded} \\ > |Tresh_i - \sum_{i=1}^k load_i| \text{Overloaded} \\ = |Tresh_i - \sum_{i=1}^k load_i| \text{Balanced} \end{cases} \tag{18}$$

Load status of a VM can be identified using the above formula, if sum of loads of all VMs are less than the threshold value of VM, then it is identified as under loaded. An under loaded VM accepts load from overloaded VM until it becomes balanced. Flowchart model for different load balancing strategy is displayed in Fig. 1.

A directed acyclic graph $W = (T, D)$ represents the workflow, and the scheduling scheme is denoted by a

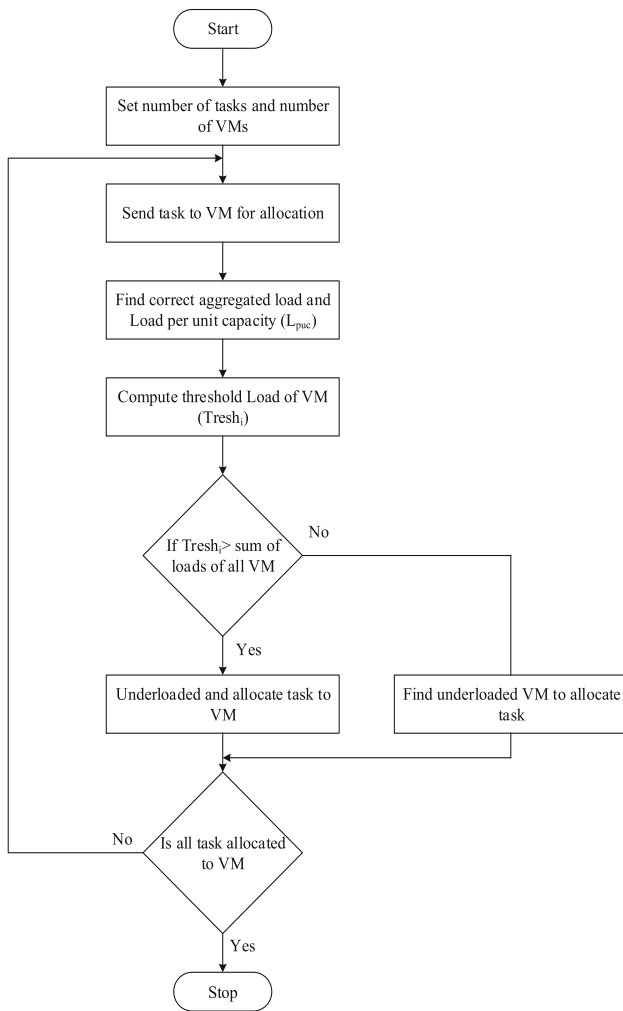


Fig. 1 Flowchart model for load balancing technique

$Schedule = (R, M, Cost, Time, Load)$. Where, load mentions the size of the task, M mentions the resource task mapping, time mentions the execution time and R denotes the resources. Here the problem is to design workflow schedules that determine the task scheduling order. Task scheduling order must comply with the dependency requirements between undertaking tasks. The task cannot be scheduled before each of its predecessors is scheduled [32, 33]. The issues can be characterized as finding a schedule with least execution cost, execution time and minimize task migration for load maintenance in scheduling. The security of data while transfer in scheduling is a significant requirement and needs to extend security measures in the proposed work.

3.5 Multi-Objective Optimization Problem (MOP)

The MOP has various conflicting objectives that should be optimized concurrently. In that view, workflow scheduling is also a typical MOP where conflicting objectives like

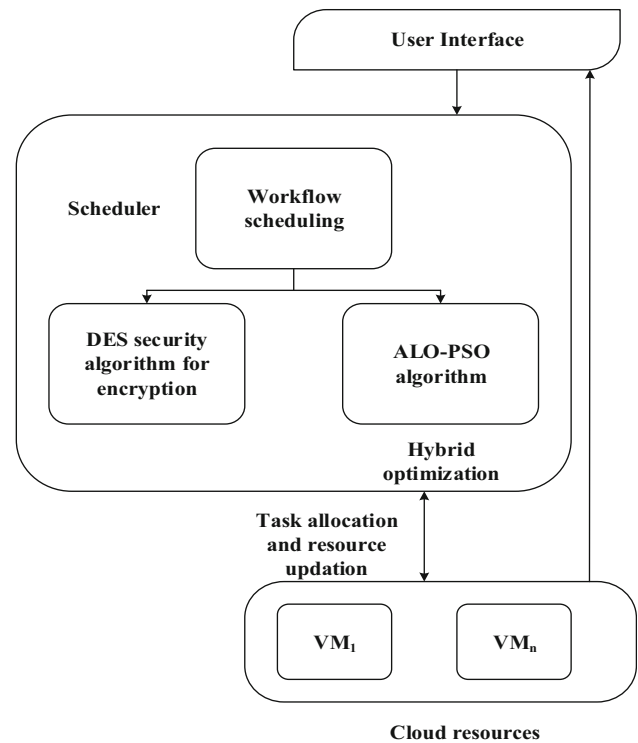


Fig. 2 Scheduling model of proposed system

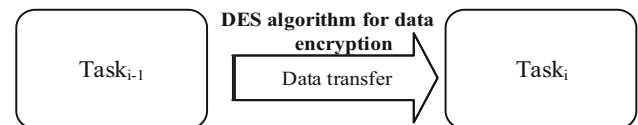


Fig. 3 DES in workflow scheduling

makespan, cost, and load are to be optimized. Multi-objective optimization problems with many decision variables and objectives can be formally defined as,

$$Fitness\ Function\ F(x) = \min(\omega_1 f_1(x) + \omega_2 f_2(x) + \omega_3 f_3(x)) \tag{19}$$

Here, makespan, cost and load balancing rate are some metrics taken to perform the optimization process. In our work, these three metrics are taken to reduce a whole objective function. First objective function is makespan. The expression for makespan is given as below:

$$f_1(x) = Makespan \tag{20}$$

Here, the equation for makespan is already displayed in Eq. (13). The second objective is the cost minimization, and it is expressed in a given below expression.

$$f_2(x) = Cost \tag{21}$$

The cost function is mentioned in Eq. (14). Finally, the last objective is load balancing rate which is expressed in the given below expression.

$$f_3(x) = Ld \quad (22)$$

Here also, the mathematical term of load balancing rate is mentioned in Eq. (15).

4 Proposed methodology

The issues indicated in Sect. 3 are NP-hard and a meta-heuristic approach could only provide alternative solutions. In this paper, a novel multi-objective hybrid algorithm to improve the workflow scheduling condition is proposed. These algorithms can perform multi-objective optimization and solve NP-hard issues. The workflow schedule is improved by utilizing the multi-objective hybrid ALO-PSO. Improvement of workflow scheduling is done as far as cost, makespan, and load. A security-based workflow scheduling is utilized to offer security to scheduling, which

4.1 Security strategy

Security in the cloud mechanism is one of the fundamental concerns while handling the scheduling and other cloud-related procedures. Data alterations, the attempt of information capture are the sorts of attacks that may happen during scheduling. In this experimentation, the DES technique [34] is utilized to provide security to workflow scheduling. The DES works dependent on a similar key to encode and decode a system of procedures. This depends on two essential properties of cryptography called substitution and transposition.

The DES comprises 16 steps, every one of which is called a round. The DES algorithm gives security to the data and the procedures are demonstrated in Fig. 3. The data are encoded and moved while scheduling is carried out. These will counteract the data leakage and gives better security.

Algorithm 1: Data Encryption Standard

Input: 64 bit plain data
Output: 64 bit cipher data
 Stage 1: Plain data of 64 bit gave the starting function of permutation.
 Stage 2: Carried out the initial permutation on plain data.
 Stage 3: Creating dual halves in permuted block called left plain text (LPT) and right plain text (RPT).
 Stage 4: The 16 rounds of encryption process is carried out on each LPT and RPT.
 Stage 5: Re-joining the LPT and RPT.
 Stage 6: Performing the final permutation on the combined block.

counteracts the alteration and loss of information. The datasets are created manually by DAG, and the execution time matrix as well as communication matrix are considered for the experimentation. The input data are inserted into the hybrid algorithm, which provides the optimized results. The security procedures are implemented by utilizing a data encryption scheme (DES) mechanism.

The process flow of proposed methodology is illustrated in Fig. 2. Experimental simulation is demonstrated in the CloudSim toolbox and the outcomes were contrasted with existing algorithms. The proposed multi-objective enhancement procedure and security measure give a successful trade-off between objectives by improved outcomes. The ideal results will be obtained from the experimentation regarding cost, makespan, and load balancing.

The security features implemented during workflow scheduling will avoid the data leakage and improves the data confidentiality. DES algorithm helps to decrypt the data while task scheduling is carried out [35]. Algorithm 1 represents the DES algorithm.

4.2 Ant-lion optimization

ALO considered as a novel nature-inspired method, which imitates a foraging procedure of ant-lions. The ant's travelling turns out to be random when searching for the food, to model such movement, the random walk is determined as,

$$X(t) = [0, \text{cumulsum}(2R(t_1) - 1), \text{cumulsum}(2R(t_2) - 1), \dots, \text{cumulsum}(2R(t_n) - 1)] \quad (23)$$

Here, n is the maximum number of iteration, t represents the iteration, $R(t)$ is the stochastic function defined as below,

$$R(t) = \begin{cases} 1, & \text{if } rand > 0.5 \\ 0, & \text{if } rand \leq 0.5 \end{cases} \quad (24)$$

where $rand$ is the random number created in the period of $[0, 1][0, 1]$. Ant’s positions are stored in the matrix for optimization.

$$Matrix_{Ant} = \begin{bmatrix} Ant_{1,1} & Ant_{1,2} & \cdots & Ant_{1,d} \\ \dots & \vdots & \dots & \vdots \\ Ant_{n,1} & Ant_{n,2} & \cdots & Ant_{n,d} \end{bmatrix} \quad (25)$$

where $Matrix_{Ant}$ is the matrix having position of each ant, n be the number of ants, and d is the number of variables. The position of ant represents parameter for a particular solution. Fitness function is utilized in optimization procedure to evaluate each ant, and the fitness value obtained thus is stored in the matrix as follows,

$$Matrix_{Fitness} = \begin{bmatrix} f(Ant_{1,1}, Ant_{1,2}, \dots, Ant_{1,d}) \\ \vdots & \dots & \dots \\ f(Ant_{n,1}, Ant_{n,2}, \dots, Ant_{n,d}) \end{bmatrix} \quad (26)$$

Similar to the case of ants, position and fitness value of the ant lions is also stored as $Matrix_{antlion}$ and $Matrix_{fital}$ respectively.

4.2.1 Ant’s random walks

The random walks depend on Eq. (23), and the position is updated in every optimization step with random walks. Given below expression is utilized to limit the ant’s random walks inside the search space.

$$X_i^t = \frac{(X_i^t - A_i) \times (D_i - C_i)}{(D_i - A_i)} + c_i \quad (27)$$

In Eq. (27), A_i is i th variable’s minimum random walk, i th variable’s maximum random walk is denoted as C_i is C_i^t gives minimum of t th iteration value for i th variable and D_i^t is a maximum of t th iteration value for i th variable.

4.2.2 Entrapment of ants in pits

Ant’s random walks are influenced by traps of ant-lions. For the mathematical representation of this condition, the below equations are used,

$$C_i^t = C^t + Antlion_j^t \quad (28)$$

$$D_i^t = D^t + Antlion_j^t \quad (29)$$

where C^t and D^t be the minimum and maximum of all variables at t th iteration. $Antlion_j^t$ is a position of j th ant-lion

during t th iteration. Roulette wheel concept is applied to trap an ant by ant-lion. During optimization procedure, the roulette wheel selects fittest ant-lion for trapping.

4.2.3 Hunting prey and re-creating pits

Ant-lion catches the ant when the fitness value of ant becomes higher than the related ant-lion. Then ant-lion needs to update its location to the current location of a hunted ant. This action increases the possibility of hunting another ant.

$$Antlion_j^t = Ant_i^t \text{ if } f(Ant_i^t) > f(Antlion_j^t) \quad (30)$$

4.2.4 Elitism

This allows maintaining the relevant solution in an optimization stage. The correct ant-lion attained so far in every iteration has stored and termed as elite. The position of ant has updated through the random walk nearby an ant-lion chosen through roulette wheel, and random walk around an elite as given below

$$Ant_i^t = \frac{RW_A^t + RW_E^t}{2} \quad (31)$$

where Ant_i^t is the position of i th ant at t th iteration. RW_A^t , is the random walk around ant-lion taken by roulette wheel at t th iteration RW_E^t , is the random walk around the elite at t th iteration [36].

4.3 Particle swarm optimization

PSO is a stochastic optimization technique based on the behavior of swarm of animals. Here, particles are the basic elements that can traverse through the problem space. This movement of particles in the problem space forms the result for the problem behind optimization. The position and velocity are the particle attributes. The optimum position achieved for each particle $pbest_i$ global best position attained for any of the particles of the whole swarm $gbest_i$ are used to calculate the velocity of a particle in every single iteration of the algorithm. Fitness function calculates the goodness of a particle’s position at each step. Depends on the fitness function, the velocity of every single particle is upgraded to a best locations, and global best positions. Velocity and location of a particles are changed in terms of Eqs. (32) and (33), respectively [37]. The algorithm continues to work until a stopping criteria is satisfied. Movement of particles towards the best position and global best positions are controlled by random numbers.

$$\begin{aligned}
 vel_i(t+1) = & IW(t)vel_i(t) + \phi_1 r_1 (pbest_i - X_i(t)) \\
 & + \phi_1 r_2 (gbest - X_i(t))
 \end{aligned}
 \quad (32)$$

Velocity equation shows different parameters that influence the convergence of algorithms such as $IW(t)$, ϕ_1 and ϕ_2 . In the equation, t be the iteration number and $IW(t)$ is the inertia weight of cycle t . Then $pbest_i$ and $gbest_i$ are the best location of a particle i and the global best location of entire particles. r_1 and r_2 are two random numbers related to ϕ_1 and ϕ_2 . Where ϕ_1 is the learning factor and ϕ_2 is the social factor and also, it fulfils a condition $\phi_1 + \phi_2 \geq 4$ [38]. Utilizing the velocity equation, the position of the particle is updated,

$$X_i(t+1) = X_i(t) + vel_i(t+1) \quad (33)$$

4.4 Workflow scheduling using hybrid ALO and PSO

Finding a technique to decide the type and number of VM is the initial step to clear the problem. The on-demand VMs are costlier when compared to the pre-allocated ones, and the algorithm preferred to utilize allocated VMs. Here, predefined sets of VMs are configured to prepare scheduling information. The different initial set of resources can be used to find various solutions to achieve the scheduling objective. Only a single pricing method is used for a single schedule. Hence pricing standards of cloud suppliers do not have effects on cost. Tasks in the workflow are assigned by the scheduler to cloud VM. The scheduler distinguishes the execution time and refreshes the task

history database. The step by step procedure of scheduler is displayed in Algorithm 2.

Algorithm 2: Scheduler

Scheduler Algorithm

- Step 1: Create a ready task list, ($task_i$)
- Step 2: Determine the idle time of each VM in the future
- Step 3: If ($ready_task_size > 0$)
- Step 4: Gather optimized value from optimizer for suitable tasks
- Step 5: Send ready tasks to VM
- Step 6: Update ready task list
- Step 7: Repeat steps 1 to 6 until the tasks are in the workflow.

The scheduler manages the optimization algorithm and controls the processing. Moreover, the scheduler is responsible for the termination of workflows. In cloud-based framework, different strategies and methodologies are used for workflow scheduling process [17]. Various types of algorithms were used in past research works. In present work, it utilizes a hybrid ALO and PSO for the powerful scheduling frameworks. The PSO is a stochastic method based on group cooperation and simulates the conduct of birds foraging. The ant-lion gathering is one of the vital parts of the ALO methodologies. Proposed hybrid algorithm is presented in Algorithm 3.

The proposed system has characteristics of both ALO and PSO algorithms. The ant-lions with the capacity of better correspondence and memory can advance faster toward the ideal arrangement. In the hybrid algorithm, the search qualities of ALO are kept, and the communication characteristic benefit of PSO is embedded. During the

Algorithm 3: Proposed Hybrid Algorithm

Hybrid ALO-PSO Algorithm

- Step 1: Initialization. Randomly initialize the positions of ants as well as ant-lions.
- Step 2: Estimate ant's and ant-lion's fitness.
- Step 3: Recognize the ant-lion whose fitness is ideal and mention it as elite.
- Step 4: Elect an ant-lion using a Roulette wheel and apply expressions 28 and 29 to update c and d .
- Step 5: Calculate a random walk around a selected ant-lion and elite, normalize using equation 23 and 27.
- Step 6: Update ant position using equation 31.
- Step 7: Repeat steps 4 to 6 for all the ants.
- Step 8: Evaluate the fitness of entire ants and change the position of ant-lion with the corresponding ant when the fitness value of ant becomes better. (Equation 30)
- Step 9: PSO is embraced to search for better ant-lions and update positions with equation 33.
- Step 10: Modify the elite when an ant lion became adequate compared to the previous elite.
- Step 11: Reiterate steps 4 to 10 while loop conditions are met.

workflow scheduling, the parameters like cost, makespan, and load are optimized in the proposed method. The ALO strategy forms the scheduling systems, and the PSO calculations help to search the better ant-lions in the ALO algorithm. This combination improves overall optimization mechanisms.

5 Simulation outcomes and discussions

In this section, simulation outcomes and the execution details of proposed system are discussed. CloudSim toolbox is utilized for cloud simulation purposes. It is an extensible simulation framework that empowers consistent modelling and simulation of cloud resources and applications. The workflow scheduling is optimized in terms of cost, makespan and load, and also provides security. The simulations operated on the dataset, which is created with 100 tasks. Communication and execution time of tasks are considered for the experimentation. The tests were kept running on an Intel i7 quad-core processor with 12 GB of DDR4 RAM. This section presents the test arrangement and examines the test results. The scheduling scheme is enhanced with the proposed hybrid algorithm. The data security is guaranteed by using DES calculations.

5.1 Task representation

The best way to represent tasks of a workflow application is in the form of DAG i.e., $G = (T, D)$ where T denotes n number of vertices or tasks i.e., $T = \{T_0, T_1, \dots, T_n\}$ and D denotes set of edges. Edges are used to measure the transfer delay time between any two different tasks i.e. T_i and T_j . It also represents dependency constraint among tasks. For suppose T_j cannot start unless task T_i completes its execution. Tasks without any predecessor and successor are termed as entry (T_{entry}) and exit (T_{exit}) tasks respectively. If the DAG consists of more than one (T_{entry}) and (T_{exit}) then a new entry and exit pseudo task is appended with zero transfer delay and computation time. All (T_{entry}) and (T_{exit}) tasks are connected with the newly created pseudo task [16]. A sample DAG with 10 tasks is shown in the Fig. 4.

In this experimentation, the processing is carried out with 100 tasks. The task properties and relation between the tasks are taken for the evaluation.

The characteristics of VMs which are utilized in the workflow scheduling process are indicated in Table 2. Basically, five VMs with 512 Mb memory is considered for execution purpose. The usage of VM brings down the number of hardware and limits the processing costs.

The properties of tasks used in the execution of workflows are indicated in Table 3. In this work, the 100 tasks

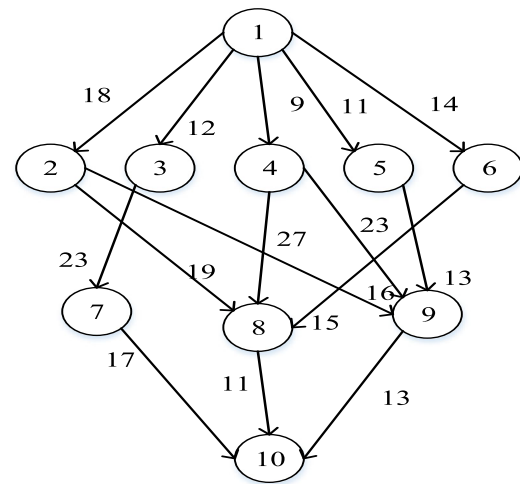


Fig. 4 Sample DAG structure for 10 tasks

are taken with various lengths. The file size, output size, and required CPU of only 30 tasks are indicated in the tabulation.

5.2 Comparison of proposed algorithm with existing methods

To distinguish the viability of proposed system, five different algorithms are examined, and their outcomes are compared. The tests had been done to a similar test bed, and the correlation is consistent. The round-robin scheduling, ALO, PSO and hybrid GA-PSO, GA [19] algorithms are taken for contrasting with the proposed ALO-PSO methods. In this experimentation firstly the values of optimization objectives makespan, cost and load are obtained and compared with the above mentioned existing methods. Proper load balancing with less task migration and reduced makespan enhances the overall performance. To ascertain this energy consumption and system reliability values are also examined. For the calculation of these values, energy and reliability models offered in the literature [29] are followed which is discussed in Sect. 3.4. The existing scheduling methods are implemented as per the algorithms available in the literature. The ant-lion and particle swarm enhancement algorithms were examined in the above section. The remaining procedure utilized for scheduling is round-robin (RR). RR scheduling depends on time-sharing, giving each task a scheduled time. The scheduling of workflows depends on the timeslots, and the scheduler chooses the task in the ready queue to execute. These are considered as a pre-emptive technique. The comparative results of proposed and existing optimization procedures are demonstrated by means of cost, makespan, system reliability, load balancing rate, as well as energy consumption.

Table 2 Characteristics of VM

VM Id	MIPS	VM image size MB	VM memory MB	Bandwidth Mbps	No. CPUs	VM name
0	250	10,000	512	1000	1	Xen
1	250	10,000	512	1000	1	Xen
2	250	10,000	512	1000	1	Xen
3	250	10,000	512	1000	1	Xen
4	250	10,000	512	1000	1	Xen

Table 3 Task properties

Task ID	Length (MIPS)	Size of file (kB)	Size of output (kB)	CPUs required
0	623,724	300	300	1
1	488,852	300	300	1
2	517,547	300	300	1
3	284,240	300	300	1
4	190,872	300	300	1
5	381,671	300	300	1
6	448,171	300	300	1
7	356,685	300	300	1
8	801,082	300	300	1
9	292,754	300	300	1
10	113,514	300	300	1
11	136,230	300	300	1
12	136,210	300	300	1
13	307,347	300	300	1
14	540,053	300	300	1
15	360,222	300	300	1
16	474,948	300	300	1
17	725,592	300	300	1
18	57,562	300	300	1
19	355,705	300	300	1
20	567,784	300	300	1
21	402,363	300	300	1
22	142,229	300	300	1
23	510,173	300	300	1
24	231,496	300	300	1
25	401,946	300	300	1
26	149,701	300	300	1
27	274,133	300	300	1
28	186,888	300	300	1
29	586,383	300	300	1

5.2.1 Cost evaluation

Execution cost of various optimization algorithms for varying number of tasks are indicated in Table 4. The cost is calculated by considering the costs of data centre, computing capacity, memory, bandwidth, storage of each VM instance type. A particular cost for each type is taken in the simulation, and the total cost is calculated by

summing each task processing cost. Results show that the proposed ALPSO achieved lowest cost, this attributed to the minimal utilization of resources by hybrid algorithm through proper allocation of tasks. Also, it selects appropriate VMs to complete least implementation cost to implement all the tasks.

Figure 5, cost analysis point outs the lowest execution cost of the proposed ALPSO. When large numbers of VMs

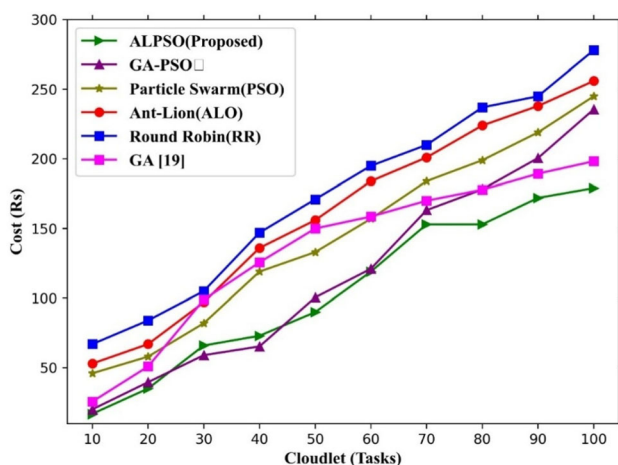
Table 4 Cost evaluation

Cloudlet (tasks)	Cost (Rs)					
	ALPSO (proposed)	GA-PSO	Particle swarm (PSO)	Ant-Lion (ALO)	Round Robin (RR)	GA [19]
10	16.8	20.03	46.0	53.0	67.0	25.6
20	35.0	39.58	58.0	67.0	84.0	50.88
30	66.0	59.04	82.0	97.0	105.0	98.89
40	73.0	65.32	119.0	136.0	147.0	125.65
50	90.0	100.51	133.0	156.0	171.0	150
60	119.0	121.05	157.0	184.0	195.0	158.6
70	153.0	163.2	184.0	201.0	210.0	170.00
80	153.0	178.11	199.0	224.0	237.0	177.86
90	172.0	200.56	219.0	238.0	245.0	189.35
100	179.0	235.58	245.0	256.0	278.0	198.25

are available, probabilities of picking appropriate VMs are high due to the extensive search ability of the proposed hybrid algorithm. The upsurge in the number of task increases a cost of scheduling. Existing algorithms consume more cost than the proposed hybrid algorithm. From this graph, it can be identified that the proposed algorithm optimizes overall cost. The other observations from the cost values are, the deterministic algorithm RR consumes more cost than the considered stochastic algorithms. Hybrid algorithms give better results than pure algorithms. For example, PSO and ALO give cost value 245 and 256 respectively for 100 tasks, while the hybrid GAPSO and ALPSO incur only 235.58 and 179 respectively.

5.2.2 Makespan evaluation

The total length of schedule until all the tasks are finished is termed as makespan. Makespan is usually distinguished

**Fig. 5** Cost analysis of existing and proposed algorithms

by assessing the time difference between the beginning and end time of a schedule [39]. The makespan of workflow is calculated using Eqs. (12) and (13).

The makespan values of optimization algorithms for varying number of tasks are given in Table 5. Results show that the proposed ALPSO achieved the lowest makespan value than other compared algorithms. ALPSO presents an improvement of 8%, 10%, 20%, 35% and 45% in comparison with GAPSO, PSO, ALO, RR and GA respectively in makespan value. This improvement is due to better search ability and convergence capacity of proposed algorithm.

Figure 6 demonstrates the performance of proposed ALPSO compared to other algorithms in makespan analysis. The deterministic algorithm RR and GA among metaheuristic algorithms take more time to execute all tasks. Proposed algorithm shows a consistent performance for all number of tasks. It is also worth to note that, hybridization of algorithms significantly improves the makespan. The ant-lion algorithm and particle swarm algorithm, when taken alone, shows higher makespan value than the proposed one.

5.2.3 Load balance evaluation

Load balancing procedures move tasks to under-loaded VMs for an efficient workflow scheduling. In cloud computing, distinctive load balancing techniques are given. The efficiency of the optimization algorithm lies in the ability to place tasks to the most suitable VMs at the initial phase itself. This simulation pursues the task migration process when over-loading occurs in VMs. The over-loaded tasks are moved to another VM. Our proposed algorithm tries to minimize task migration for better scheduling. Task migration can be minimized by limiting the load on VMs.

Table 5 Makespan

Cloudlet	Makespan (s)					
	ALPSO (proposed)	GA-PSO	PSO	ALO	RR	GA [19]
10	16.9	68.44	44.9	21.9	53.9	50.23
20	34.08	95.09	63.8	45.8	74.8	85
30	58.7	100.54	89.7	79.7	120.7	90.25
40	70.6	110.8	139.6	101.6	159.6	200.5
50	86.6	116.01	157.6	136.6	169.6	250
60	114.5	125.05	178.5	152.5	187.5	288.36
70	146.5	160.58	205.4	174.3	220.5	300.25
80	164.4	191.02	235.4	185.4	249.5	325.5
90	182.5	223.5	253.4	223.3	268.5	357.7
100	214.3	233.78	264.4	253.3	276.5	380

Load balancing will maximize the throughput and minimize the reaction time [40].

Load balancing values of proposed and existing optimization algorithms are indicated in Table 6. Our efficient optimization algorithm provides less overloading. By using PSO algorithm, the ALPSO method converges to the solution in a better way to avoid an unnecessary diversity which may worsen the algorithm quality. Non-efficient algorithm provides more loads to VMs [42].

Performance comparison of load balancing with a various task is indicated in Fig. 7. When tasks increases naturally, overhead in VMs also increases. The task migration will also increase, which is clear from Fig. 7. The overloaded tasks are then migrated to a new VM and the loads are balanced. From the results, it is clear that better load optimization is possible in the proposed hybrid method. The GA scheduling algorithm shows the highest possibilities of VM overloading and hence more number of migrations. But for more number of tasks GA shows a comparatively good load balancing rate. For example as shown in Table 6 load balancing rate of GA for 20 tasks is

52.8 where as it is only 49.2 for 100 tasks. However our proposed algorithm exhibits a consistent performance for all number of tasks.

5.2.4 Energy consumption evaluation

Total energy consumed by the computing resources which implements all tasks of the workflow application is termed as energy consumption. Table 7 provides the comparison values for energy consumption of existing algorithms.

Evaluation of energy consumption is displayed in Fig. 8. The performance is compared with previous methods like reliability and energy efficient workflow scheduling (REEWS), HEFT, Power Aware List-based Scheduling (PALS), and RHEFT [29] methods. The REEWS and PALS algorithms which are focusing on energy parameter consume less energy compared to HEFT and RHEFT. Here the energy consumption rate of the proposed method is as good as REEWS and it outperforms PALS.

5.2.5 Reliability evaluation

Table 8 provides the comparison values for system reliability for existing methods. Probability of execution of all number of tasks without any failure is defined as the reliability of the system.

Figure 9 displays the evaluation of system reliability. The results show the good reliability performance of the solution achieved by the proposed method. The REEWS algorithm which is specifically designed for reliability and energy exhibits high reliability. But, in case of PALS, reliability is small because this focuses on the optimization of execution time and energy consumption. Reliability of both HEFT and RHEFT are high as they select normal frequency of the processor. From the above analysis it is evident that the reliability and energy consumption rate of the schedules created by the proposed method is

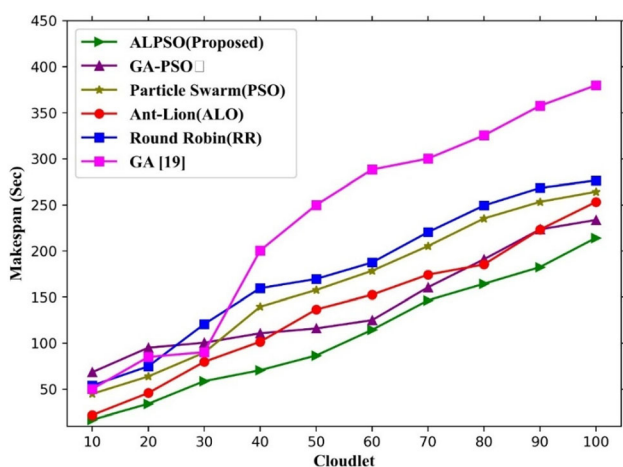
**Fig. 6** Comparison of makespan

Table 6 Load balancing in scheduling

Cloudlet	Load balancing rate					
	ALPSO (proposed)	GA-PSO	ALO	PSO	RR	GA [19]
10	2.0	7.58	4.0	5.0	6.0	20.6
20	5.0	9.76	7.0	8.0	12.0	52.58
30	8.0	10.65	10.0	11.0	13.0	55.6
40	10.0	12.7	12.0	14.0	18.0	54.2
50	13.0	13.81	15.0	17.0	20.0	61.93
60	16.0	17.55	18.0	19.0	23.0	58.5
70	18.0	21.02	20.0	22.0	24.0	40.5
80	21.0	27.8	23.0	24.0	27.0	42.58
90	24.0	30.25	26.0	27.0	29.0	45.56
100	27.0	33.03	29.0	31.0	35.0	49.2

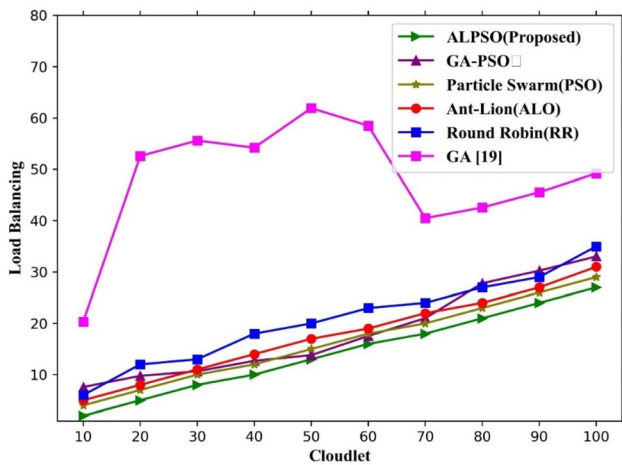


Fig. 7 Evaluation of load balancing

comparable with that of algorithms specifically designed for energy and reliability.

6 Conclusion

In past years, the researchers have focused on cloud workflow scheduling with a single objective. Nowadays, a few of the researches attempted to solve multi-objective problems. This research is predominantly intended to

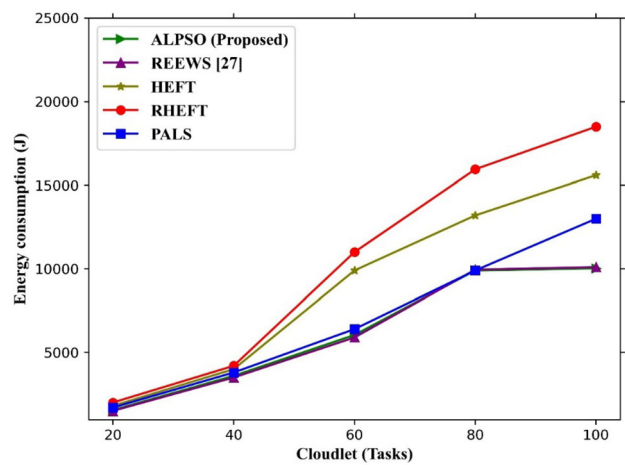


Fig. 8 Evaluation of energy consumption

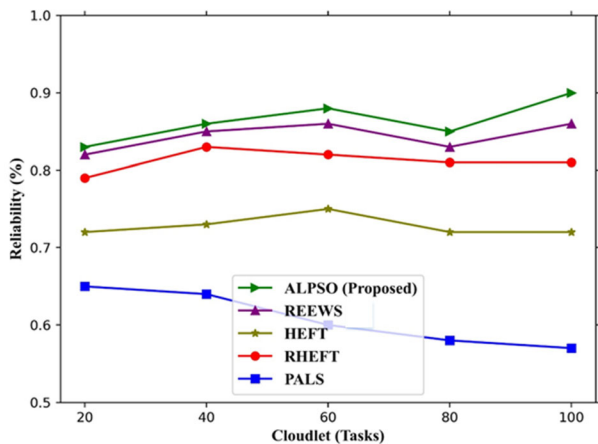
lessen cost, makespan, and a load of workflow applications on IaaS clouds in the context of workflow execution. The situation was modelled as a multi-objective optimization problem. Hybrid multi-objective ALO and PSO improvement algorithms are proposed as a solution. In the present work, the adequacy of hybrid algorithm is examined for multi-objective workflow scheduling in the cloud model. The workflow datasets are executed and assessed utilizing the CloudSim tool. The novel hybrid algorithm returned

Table 7 Comparison of energy consumption

Cloudlet (tasks)	Energy consumption				
	ALPSO (proposed)	REEWS	HEFT	RHEFT	PALS
20	1550	1500	1800	2000	1700
40	3600	3500	4000	4200	3800
60	6050	5900	9900	11,000	6400
80	9900	9950	13,200	15,950	9900
100	10,025	10,100	15,600	18,500	13,000

Table 8 Comparison of system reliability

Cloudlet (tasks)	Reliability				
	ALPSO (proposed)	REEWS	HEFT	RHEFT	PALS
20	0.83	0.82	0.72	0.79	0.65
40	0.86	0.85	0.73	0.83	0.64
60	0.88	0.86	0.75	0.82	0.60
80	0.85	0.83	0.72	0.81	0.58
100	0.90	0.86	0.72	0.81	0.57

**Fig. 9** Evaluation of system reliability

better scheduling plans that reduce cost, makespan, and load. The security procedure includes data encryption model which offers better security while scheduling. Simulations are conducted using randomly generated workflows with varying numbers of tasks. The results prove the superiority of hybrid meta-heuristic algorithms in scheduling workflows. The proposed hybrid algorithm outperforms previous meta-heuristic algorithms like GA-PSO, PSO, ALO, and deterministic algorithm RR. Simulation values show that the proposed method minimizes the cost by 9.8% of GA-PSO, 10% of PSO, 20% of ALO, 30% of RR and 12% of GA. Load balancing and makespan objectives of the proposed method enhance 8% than GA-PSO 10% than ALO, 20% than PSO, 35% than RR and 45% than GA. The energy consumption and reliability performance of the solution generated by the proposed method is also promising. High convergence rate and searchability of the ALO algorithm, as well as the communication capability of PSO are contributed significantly towards the performance of this proposed method. These strategies can be well applied to resolve diverse multi-objective enhancement issues in the workflow scheduling scenario. In future, the present work can be protracted by utilizing diverse QoS requirements like success rate, trust management and so forth to efficiently perform task

scheduling. It is also good to consider more than one pricing scheme for VM leasing in future enhancements. The options for executing and extending these strategies for multi-clouds also suggested. Multi-cloud environment integrate multiple clouds together to offer a unified service in a collaborative manner.

References

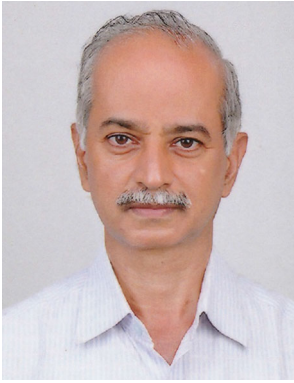
- Varghese, B., Buyya, R.: Next generation cloud computing: new trends and research directions. *Future Gen. Comput. Syst.* **79**(3), 849–861 (2018)
- Han, G., et al.: An efficient virtual machine consolidation scheme for multimedia cloud computing. *Sensors* **16**(2), 246 (2016)
- Aral, A., Ovatman, T.: Network-aware embedding of virtual machine clusters onto federated cloud infrastructure. *J. Syst. Softw.* **120**, 89–104 (2016)
- Ramachandran, M., Chang, V.: Towards performance evaluation of cloud service providers for cloud data security. *Int. J. Inf. Manage.* **36**, 618–625 (2016)
- Amato, F., Moscato, F.: Exploiting cloud and workflow patterns for the analysis of composite cloud services. *Future Gen. Comput. Syst.* **67**, 255–265 (2017)
- Prathyusha, J., Sandhya, G., Reddy, V.K.: An improvised partition-based workflow scheduling algorithm. *Int. J. Pure Appl. Math.* **115**(7), 381–385 (2017)
- Chirkin, A.M., et al.: Execution time estimation for workflow scheduling. *Future Gen. Comput. Syst.* **75**, 376–387 (2017)
- Ghahramani, M.H., Zhou, M., Hon, C.T.: Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services. *IEEE/CAA J. Automat. Sin.* **4**(1), 6–18 (2017)
- Jouini, M., ArfaRabai, L.B.: A security framework for secure cloud computing environments. *Int. J. Cloud Appl. Comput.* **6**(3), 249–263 (2016)
- Bhushan, K., Gupta, B.B.: Security challenges in cloud computing: state-of-art. *Int. J. Big Data Intell.* **4**(2), 81–107 (2017)
- Rahi, S.B., Bisui, S., Misra, S.C.: Identifying critical challenges in the adoption of cloud-based services. *Int. J. Commun. Syst.* **30**(12), e3261 (2017)
- Hudic, A., Smith, P., Weippl, E.R.: Security assurance assessment methodology for hybrid clouds. *Comput. Secur.* **70**, 723–743 (2017)
- Kaur, P., Mehta, M.: Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *J. Parallel Distrib. Comput.* **101**, 41–50 (2017)
- Mishra, S.K., Manjula, R.: A meta-heuristic based multi objective optimization for load distribution in cloud data center under varying workloads. *Clust. Comput.* **19**, 1–5 (2020)

15. Souri, A., Rahmani, A.M., Navimipour, N.J., Rezaei, R.: A hybrid formal verification approach for QoS-aware multi-cloud service composition. *Clust. Comput.* **28**, 1–8 (2019)
16. Verma, A., Kaushal, S.: A hybrid multi-objective particle swarm optimization for scientific workflow scheduling. *Parallel Comput.* **62**, 1–19 (2017)
17. Casas, I., et al.: A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems. *Future Gen. Comput. Syst.* **74**, 168–178 (2017)
18. Choudhary, A., et al.: A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing. *Future Gen. Comput. Syst.* **83**, 14–26 (2018)
19. Manasrah, A.M., Ali, H.B.: Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. *Wireless Commun. Mobile Comput.* **2018**, 1–16 (2018)
20. Pang, S., Li, W., He, H., Shan, Z., Wang, X.: An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing. *IEEE Access.* **7**, 146379–146389 (2019)
21. Chen, Z.-G., et al.: Multi objective cloud workflow scheduling: a multiple population ant colony system approach. *IEEE Trans. Cybern.* **49**(8), 2912–2926 (2019)
22. Rehman, A., et al.: Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurrency Comput. Pract. Exp.* **31**(8), e4949 (2018)
23. Shishido, H.Y., et al.: Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. *Comput. Electrical Eng.* **69**, 378–394 (2018)
24. Sujana, A.J., Revathi, T., Rajanayagam, S.J.: Fuzzy-based security-driven optimistic scheduling of scientific workflows in cloud computing. *IETE J. Res.* **66**, 224–241 (2018)
25. Sharma, C., Rashid, M.: Scheduling of Scientific Workflow in Distributed Cloud Environment Using Hybrid PSO Algorithm. In *Trends in Cloud-based IoT*, pp. 113–123. Springer, Cham (2020)
26. Iranmanesh, A., Naji, H.R.: DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing. *Cluster Comput.* (2020). <https://doi.org/10.1007/s10586-020-03145-8>
27. Abualigah, L., Diabat, A.: A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Cluster Comput.* **12**, 1–9 (2020)
28. Ghobaei-Arani, M., Souri, A., Safara, F., Norouzi, M.: An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing. *Trans. Emerg. Telecommun. Technol.* **31**(2), e3770 (2020)
29. Garg, R., Mittal, M.: Reliability and energy efficient workflow scheduling in cloud environment. *Cluster Comput.* **22**(4), 1283–1297 (2019)
30. Keshanchi, B., Souri, A., Navimipour, N.J.: 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 (2017)
31. Zhou, X., et al.: Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. *Future Gen. Comput. Syst.* **93**, 278–289 (2019)
32. Zhu, Z., et al.: Evolutionary multi-objective workflow scheduling in cloud. *IEEE Trans. Parallel Distrib. Syst.* **27**(5), 1344–1357 (2016)
33. Arabnejad, H., Barbosa, J.G., Prodan, R.: Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources. *Future Gen. Comput. Syst.* **55**, 29–40 (2016)
34. Oukili, S., Bri, S.: High throughput FPGA Implementation of Data Encryption Standard with time variable sub-keys. *Int. J. Electrical Comput. Eng.* **6**(1), 298–306 (2016)
35. Arboleda, E.R., Balaba, J.L., Espineli, J.C.L.: Chaotic Rivest-Shamir-Adleman algorithm with data encryption standard scheduling. *Bull. Electrical Eng. Inf.* **6**(3), 219–227 (2017)
36. Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015)
37. Du, K.-L., Swamy, M.N.S.: Particle swarm optimization Search and optimization by metaheuristics, pp. 158–173. Birkhäuser, Cham (2016)
38. Zhang, C., et al.: Particle swarm optimization algorithm based on ontology model to support cloud computing applications. *J. Ambient Intell. Hum. Comput.* **7**(5), 633–638 (2016)
39. Abdullahi, M., Ngadi, M.A., Abdulhamid, S.M.: Symbiotic Organism Search optimization based task scheduling in cloud computing environment. *Future Gen. Comput. Syst.* **56**, 640–650 (2016)
40. Ramezani, F., Lu, J., Hussai, F.K.: Task-based system load balancing in cloud computing using particle swarm optimization. *Int. J. Parallel Programm.* **42**, 739–754 (2014)
41. Farrag, A.A.S., Mohamad, S.A., El Sayed, M.: Swarm Intelligent Algorithms for solving load balancing in cloud computing. *Egypt. Comput. Sci. J.* **43**(1), 45–57 (2019)
42. Devi, D.C., Uthiaraj, V.R.: Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks. *Sci. World J.* (2016). <https://doi.org/10.1155/2016/3896065>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

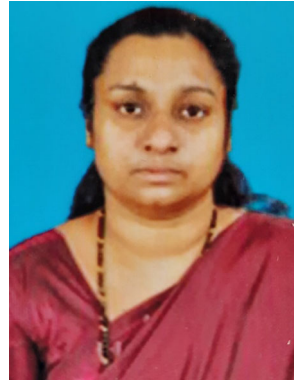


Jabir Kakkottakath Valappil Thekkepurayil is a doctoral student at School of Engineering, Cochin University of Science and Technology (CUSAT), Kochi, India. He got his masters Degree in Software engineering from the same University. His research studies include cloud computing and workflow scheduling.



David Peter Suseelan is currently working as Professor, School Of Engineering, Cochin University of Science and Technology. He received the M.Tech degree in computer science from IIT Madras, Madras, and the Ph.D. degree in computer science from the Cochin University of Science and Technology (CUSAT), Kochi, India, in 2006. He has got almost 30 years of teaching experience. He was formerly the Registrar of Cochin University

of Science and Technology. He served University in various positions like Director of International Relations and Academic Admissions, Principal of the School of Engineering and Systems Director of the Centre for Information and Resource Management. His areas of interest include artificial intelligence, software engineering and cloud computing.



Preetha Mathew Keerikkattil is a Professor in computer science and engineering at Cochin University College of Engineering, Alappuzha, Kerala, India. She received the M.Tech. degree in computer science from Cochin University of Science and Technology (CUSAT), Kochi, India and the Ph.D. degree in computer science from IIT Madras, Madras. She was formerly the Principal of Cochin University College of Engineering, Alappuzha, Kerala, India.

She has authored several research papers in various national and international journals. Her areas of interest include cryptography, Algorithms and cloud computing.