e-ISSN : 0976-5166
p-ISSN : 2231-3850

TASK SCHEDULING IN CLOUD USING RESISTIVE AND RANKING META-HEURISTIC OPTIMIZATION APPROACH

Rajkumar Kalimuthu

Research Scholar, Department of Computer Science Engineering, Noorul Islam Centre for Higher Education, Tamil Nadu, India
rajkumarengg2020@gmail.com

Brindha Thomas

Associate Professor, Department of Information Technology, Noorul Islam Centre for Higher Education, Tamil Nadu, India
brindha@niuniv.com

Abstract

Cloud computing (CC) is an emergent and revolutionary technological paradigm to facilitate data generation over the networking environment. It is determined as the computing edge where it provides cloud services over the networking point. It assists in handling the delay issues during the task scheduling process. With the advent analysis over the existing approaches, inappropriate task scheduling in cloud computing outcomes in huge delay than the other computing models. Therefore, the actual advantages of cloud computing are attained with the adoption of appropriate task scheduling strategies. However, it is an NP-hard issue and needs effectual and optimal approaches to dealing with resource utilization, response time, and latency at the networking model. This research concentrates on modelling an efficient hybrid optimization approach to overcome the drawbacks of single standard optimization approaches. This work hybridizes the resistive-based Particle swarm Optimization ($hrPSO$) approach with Improved ranking-based Grey wolf optimization ($ir - GWO$) approach to handle the exploration and exploitation issues over the standard methods. Here, $hr - PSO$ is used to schedule the task among the connected devices, and $ir - GWO$ helps manage the resources at the device level. The resources are allocated and managed based on the demand generated with the incoming requests in the anticipated model. The ultimate target of this work is to diminish the delay, average response time, and optimal resource utilization by efficiently scheduling the tasks by managing the available resources. The simulation is done with the MATLAB 2016b simulator. The evaluation results show that the anticipated model provides promising outcomes with energy consumption, average response time, execution time, etc. The proposed model shows a better trade-off in contrast to prevailing approaches.

Keywords: **Cloud computing; task allocation; resource utilization; hybrid particle swarm optimization; improved grey wolf optimization.**

1. Introduction

With the extensive growth of the Internet and the access towards big data, Cloud Computing (CC) turns to be the most popular technology all over the business world [1]. Compared to various distributed computing techniques, i.e. grid and cluster computing, CC has offered a scalable and elastic way for delivering multiple services to consumers [2]. However, consumers do not necessitate the underlying technologies, and they provide various platforms for execution and computing resources in a pay-per-use manner.

The baseline functionality of CC is to transfer computing tasks to the resource pooling constitution with the massive amount of virtualized and heterogeneous servers or virtual machines (VM) [3]. Generally, CC offers market-based efficacy, the users and cloud providers intend to maximize the investment return and profits, enhancements over the resource scheduling process that assists in user and software applications, workflows, tasks, and so on are entirely essential [4]. However, scheduling directly influences the system performance like operational cost and efficiency during resource utilization, and it is considered the supreme significance over the CC model. In general, VMs are dynamically allocated, proficient, and organized; however, the scheduling issues over the CC are partitioned into two diverse layers [5]. The initial one is the task scheduling process requested by the users and the way to map them towards the set of accessible VM resources. The second form is the host and VM mapping, which makes VM the appropriate host for the migration and the creation [6]. This work concentrates on optimizing the various problems. It influences the processing capability directly over the CC system, and optimized task scheduling has highly influenced the significance of the entire system, like computational cost and time. Moreover, the complexity of the optimization issues is determined as the NP-hard issues. It specifies that the problem-solving issues rely on exponential time, and the process suffers from the dimensionality collapse when the problem size increases.

Generally, meta-heuristic algorithms have gathered the attained of the huge researchers to deal with the complex optimization issues in an appropriate time. The cause behind this is the efficient way provided by the heuristic approaches and predicts the optimal solution approximately over polynomial-time indeed of exponential time compared to various traditional methods [7]. However, diverse meta-heuristic techniques and variations are utilized to resolve the scheduling issues in multiple fields that include CC. Based on the summary from various methods, presently, meta-heuristics approaches are being used for scheduling tasks which specifically include swarm intelligence, genetic algorithms (GA) like particle swarm optimization, and ant colony optimization [8]. These optimization approaches are attained from the execution and evolution of the population and can resolve the complex global optimization issues via competition and cooperation between the individuals [9].

The effectual resource management in the CC environment is drastically confronting owing to the dynamical characteristics of latency, computation, storage, and bandwidth of end-user devices. For instance, the records of on-duty ambulances like location, capabilities, and so on over the connected vehicle scenario and the intelligent traffic lights are managed to route the ambulance during the emergency. With certain speed and computational demands, it is essential to deal with the resources to fulfil the quality of service (QoS) requirements. Author et al. [10] anticipate the migration and placement methods like MIGCEP for resource management in both fog and cloud computing. The entire operator mitigation planning fulfils the diminished end-to-end delay and network utilization. Application-based resource provisioning assists in CC being utilized efficiently with IoT for sensing or crowdsourcing. [1]

With the extension of the cloud towards the networking edge, the efficient resource management process is considered the most challenging task [12]-[13]. However, there is an awful need to anticipate and model energy-aware and performance-based resource management approaches [14]-[15]. In this context, this research concentrates on modelling an efficient meta-heuristic hybrid optimization approach with prevailing review models. The anticipated model schedules the tasks and deals with various resources to attain effectual resource utilization and enhanced performance. Based on the expected model, the scheduler needs to predict the appropriate devices for the incoming tasks based on the CPU demand for memory and time and allocates the resource based on it. It can be achieved only when the appropriate help from the end-devices transforms the task to the cloud-based resources. The significant contributions of this research are given below:

- To hybrid the novel meta-heuristic optimization approach that merges the improved grey wolf optimization $ir - GWO$ and hybrid resistive particle swarm optimization ($hr - PSO$).

The anticipated model shows two essential contributions like resource allocation and task scheduling, over the computing devices to optimize the resource required for utilization and reduce the processing cost and response time.

- The anticipated model is simulated with MATLAB 2016b simulator, and the validation is performed with the evaluation of various prevailing approaches for resource management. The outcomes show the model's efficacy based on energy consumption, response time, and processing cost.

The remainder of the work is formulated as: In section 2, and extensive analysis is performed with the various prevailing approaches; in section 3, the anticipated system model is discussed based on the novel hybridization process with the corresponding problem statement where the proposed model needs to resolve the formulated problem. In section 4, the numerical results attained from the hybrid model are discussed, and the validation is done with the simulator model. In section 5, the conclusion with the hybridization model is shown, and the ideas for future research extension are discussed.

2. Related works

In the CC environment, task scheduling is specified in an independent manner where the flow graph for task mapping is performed through Directed Acyclic Graph. Alike of CC environment, specific optimization models are crucial for the service providers and users' expectations like QoS, energy consumption, cost, and so on. Therefore, this review concentrates on various prevailing works by determining the optimization function.

Sun et al. [16] discuss an improved Energy-aware dynamical task scheduling process over the CC environment, and the algorithm targets to reduce the total energy consumption of mobiles. It relies on the dynamical voltage scaling model by ensuring the appropriate probability constraint and time deadline for various applications. The probability constraints for specific applications helps in providing the probability of executing the task efficiently. The task scheduling approach is specified to handle the energy consumption issues over the mobile devices under the mobile CC environment. There are three diverse phases in the given algorithm. Initially, the scheduling strategies are attained using the reduced total execution time. Followed by this is a task migration process applied over the cloud tasks for migration while fulfilling the application deadline to diminish the power consumed by the mobile devices. At last, the dynamical voltage and frequency scaling model is used to reduce consumed energy when the devices meet the time deadline. The computational offloading and joint scheduling process for multi-component devices is proposed by Wang et al. [17], which leads the optimal decision for the components to offload over the cloud and orders the schedule in the offloaded components. The integer-based linear programming model is used to improve the remote execution energy.

Wang et al. [18] anticipate two diverse approaches known as the LACO and probabilistic list scheduling process. The significant objective of this work is to enhance the execution cost with the user-defined model for deadline constraints. Snapchat et al. [19] discuss an algorithm that is an improved version of the contention-based scheduling process. Initially, it specifies the exceptional order for all tasks by the assumption towards the priority. The most appropriate processor was selected to schedule the task based on the minimal trade-off between the finish time and total cost. These two problems are considered during the task scheduling process over the computing environment. The initial problem is about the task that was offloaded towards the cloud server, and the corresponding problem is related to the optimality of the server. Therefore, this work considers both the pricing models and task scheduling over the cloud service providing servers during scheduling and offloading the tasks. While considering the mobile users, the utility maximization functionality is performed during the delay, energy consumption, and cloud service price. The CSP anticipates the primal-dual approaches and convexification to facilitate the optimal pricing approaches. Sujana et al. [20] anticipate an improved genetic algorithm that reduces the trade-off solution among the execution cost and task processing time to attain superior performance.

Mohan et al. [21] consider the traditional HEFT heuristic algorithm as the task scheduling algorithm with two diverse phases. The initial phase is to choose the task with superior rank upward by determining the communication time and mean execution among the tasks to sort them randomly. Then, the processor with reduced execution time is chosen in the processor selection phase based on the insertion model to diminish the last finish time. This model attempts to attain reduced finish time despite various constraints like the QoS model. Therefore, the specific process initiates with this algorithmic model to determine the limitations. Jiang et al. [22] anticipate a budget-based heterogeneous finish time considering budget and time deadline consideration. Specific rules are constructed based on the service selection process to attain some adorable services with prior finish time. However, a heterogeneous budget-based scheduling model is used to resolve a single-objective scheduling model where the processing time needs to be optimized. The budget needs to set the constraints. Variation from the second successive phase of the model, the heterogeneous model, provides the trade-off among the finish time and budget for all processors to compute the execution process. However, both these models have the competency to resolve the constraint issues by enhancing the HEFT model; however, the author intends to focus on single-objective optimization.

Juve et al. [23] discusses the demand-based task-driven scheduling process and provides an estimation model for predicting the task finish time. This work deals with the task scheduling process from both the service providers and mobile user's perspectives. A novel 2D-chromosomal genetic algorithm relies on the weighted objective model to reduce the finish time of the end-users and attain efficient load balancing with service providers. The stochastic computation task scheduling process is anticipated by []. Here, average power consumption and the average delay are depicted at the mobile-user level with an effectual 1D searching algorithm to attain a shorter average execution delay.

Various works concentrate on the optimization objectives like load, power, QoS, etc. Silva et al. [24] anticipate a model to enhance the processor selection model with the HEFT algorithmic model. Generally, the parent principle was performed to attain QoS and load balancing while allocating tasks over the CC scenario. Similarly, the hybridized colony model for the application schedule is anticipated to resolve the scheduling process over the local cloud model. This model intends to enhance the total cost-based profit while fulfilling the resource capacity constraints for all service providers. Some work concentrates on single-objective optimization without limitations from the models mentioned above. However, Shah et al. [25] determine task scheduling as a multi-objective problem where these two multi-objective algorithms are not there to resolve the constrained issues. There are only fewer models that concentrate on constraints while determining the multi-objective concern for scheduling task over the CC environment, specifically with the extended version. Thus, this research focuses on providing a hybridized model to deal with the multi-objective issues and intends to resolve all the constraints mentioned above efficiently.

Table 1. Comparison of various existing approaches.

Algorithm	Method	Parameters	Advantages	Disadvantages
First Come First Serve (FCFS)	It helps to deal with task scheduling in FIFO queuing model, and the first-come tasks are executed over the VM first.	Arrival time measure	Faster and simple execution	Task scheduling entirely relies on arrival time; however, it does not consider any VM utilization.
Min-Min algorithm	The task shows minimal execution time for selecting all the task	Makespan	Superior makespan	Insufficient QoS Load imbalance
Particle Swarm Optimization	The model uses population to predict the optimal minimal values that assist in creating appropriate ordering tasks with the proper task.	Inertia computation and constant measure	Superior resource utilization; Prediction of optimal solution; Reducing processing time	Slow convergence speed with larger search space

Round Robin	Works with the cyclic model where the task provides an equal chance for selecting and smaller unit of time for execution	Time slice and arrival time	Superior response time;
			Balanced load;
			Less complex
				Pre-emption leads to process out, and the time slice expires

The priority-based job scheduling process	Dependency model	Queuing priority	Process prioritization increases;
			User-friendly;
			Requires essential time for resource management and less finish time
				Low priority jobs are lost with system crash
				
				Resource starvation

Greedy approach	The approach attempts to predict the global optimum for solving heuristic approaches with steps.	Domain, parameter, and population	Easier implementation;
			The algorithm requires lesser resources;
			Faster execution;
			Faster scheduling
				Global optimization is not achieved;
				Extremely complex during the process of parameter variations

Genetic algorithm	The algorithm relies on domain solution and proper functionality to evaluate the domain solution.	Population size;
		Crossover probability;
		Mutation probability
			Handles mathematical issues and financial issues are appropriate;
			Easier understanding;
			Applications need lesser time for processing
				Model is entirely slower;
				It cannot predict the exact solution
				Selection is not so appropriate

3. Methodology

This section discusses the CC model for efficiently evaluating task scheduling using meta-heuristic optimization. This work hybridizes the resistive-based Particle swarm Optimization ($hrPSO$) approach with Improved ranking-based Grey wolf optimization ($ir - GWO$) approach to handle the exploration and exploitation issues over the standard methods. Here, $hr - PSO$ is used to schedule the task among the connected devices, and $ir - GWO$ helps manage the resources at the device level. The functionality of the model is discussed in the section given below.

3.1 Problem formulation

The workflow scheduling process is based on the directed acyclic model that considers the execution independently. The scheduling process needs to optimize the cost, and the DAG should fulfil the QoS requirements. Generally, the weighted workflow is represented with DAG model $G = (T, E)$ where $'T'$ is set of $\{t_1, t_2, \dots, t_n\}$ and the workload length is given as $load(t_i)$, and $'E'$ is precedence based on the tasks, $e_{i,j}$ is precedence constraints among the task t_i and t_j. When the communication is initiated, the communication among the task needs to wait for data transfer. However, it is known as the parent task or predecessor task, and the others are known as child task. In DAG, the task without parent is known as the entry task, and the task without a child is known as exit task. The task scheduling process is based on dummy tasks like t_{entry} and t_{ex}t. The service provider offers various processing capabilities and rental costs in the cloud environment. The end-users are charged based on the time interval and uses partial utilization to specify the processing capability. While executing the task, $t_i$it is allocated and the execution time is represented as Eq. (1):

$$ET\ (t_i, I) = \frac{length\ (t_i)}{capacity} \tag{1}$$

The bandwidths of the instances are equal, and the tasks over the instances are free. The transfer time among the tasks t_i and t_j and starting time $ST(t_i)$ and $FT\ (t_i)$ as in Eq. (2) to Eq. (4):

$$Transfer\ time = \begin{cases} \frac{data}{bandwidth} & if\ i \neq j \\ 0 & else \end{cases} \tag{2}$$

$$ST\ (t_i) = \begin{cases} 0 & prediction(t_i) \neq \emptyset; \\ \max\{ST(t_j)\ ET\ (t_j, I) + TT\ (e_j EJ, i)\} & prediction(t_i) \neq \emptyset; \end{cases} \tag{3}$$

$$FT(t_i) = ST(t_i) + ET\ (t_i, I) \tag{4}$$

The completion time of the exit task is the makespan of the entire workflow, and the maximal end time of the exit task is expressed as in Eq. (5):

$$makespan = FT\ (t_{exit}) \tag{5}$$

There are two diverse issues considered during the workflow scheduling in the CC environment, i.e. actual schedule mapping or resource provisioning. The model needs a scheduling solution for determining the instances like types, rent, start and finish time. This issue concentrates on mapping the task over the appropriate resources. Generally, the schedule is expressed as $< I, A >. I = \{I_1, I_2, \dots, I_{|I|}\}$ is a set of resources and it is related with instance type where the start and end time is provided and $'A'$ is task allocation represented by $A =< t_i, s_j, ST(t_i), FT(t_i) >$ specifies task allocated to resources.

3.2 Hybrid Resistive Particle Swarm Optimization ($hrPSO$)

PSO is determined as an evolutionary model based on swarm nature in meta-heuristic optimization. It is a stochastic optimization model where it specifies the movement of the individual over the problem space and specifies candidate solutions to the optimization issues. The particle motion is determined based on vector and velocity. Thus, it possesses direction and amplitude and adopts inertia weight and constant learning. The position and speed of the particles in $'k'$ time are specified as in Eq. (6) - Eq. (7):

$$P_i^k = P_i^{k-1} + V_i^k \tag{6}$$

$$V_i^k = \omega V_i^{i-1} + c_1 r_1 (P_i^{pbest} - P_i^{k-1}) + c_2 r_2 (P_g^{gbest} - P_i^{k-1}) \tag{7}$$

Generally, the resistive system comprises molecules, cells, and organs where the system functionality relies on the non-self and stimuli recognition pretends to retain memory and accurate response. It is built with the self-regulatory mechanism where the affinity is adopted to measure the similarity between the antigen and the antibody. This model is integrated with the standard PSO model to deal with the multi-objective constraint, and it pretends to provide the candidate solution and antigens. The affinity projects the similarity among the solutions and ensures the objective function and constraint conditions. The resistivity condition relies on the system memory, where a considerable amount of antibodies are generated. PSO-based optimization is partitioned into five diverse stages. The first one is population initialization to select the particles randomly based on position and speed. Then the particles are computed for the best particle to project the optimal global particles. The next stage is the computation of velocity and condition based on the above-given formula [7], and at last, update the optimal global particles with an iterative stop condition. While processing PSO, the particles need to predict other particles more quickly. When the optimal position (current) is determined as the local best, then the computation is complex with a search space solution. However, the resistivity system does not offer any sort of unified description. The initialization and the prediction of global solutions rely on the concentric and affinity measures. The simulative value is provided for evaluating the degree of excellence. The series operation includes cloning, mutation, cross-function, cell formation and replacement, etc. Finally, the update process is done with essential stopping criteria to fulfil the basic requirements.

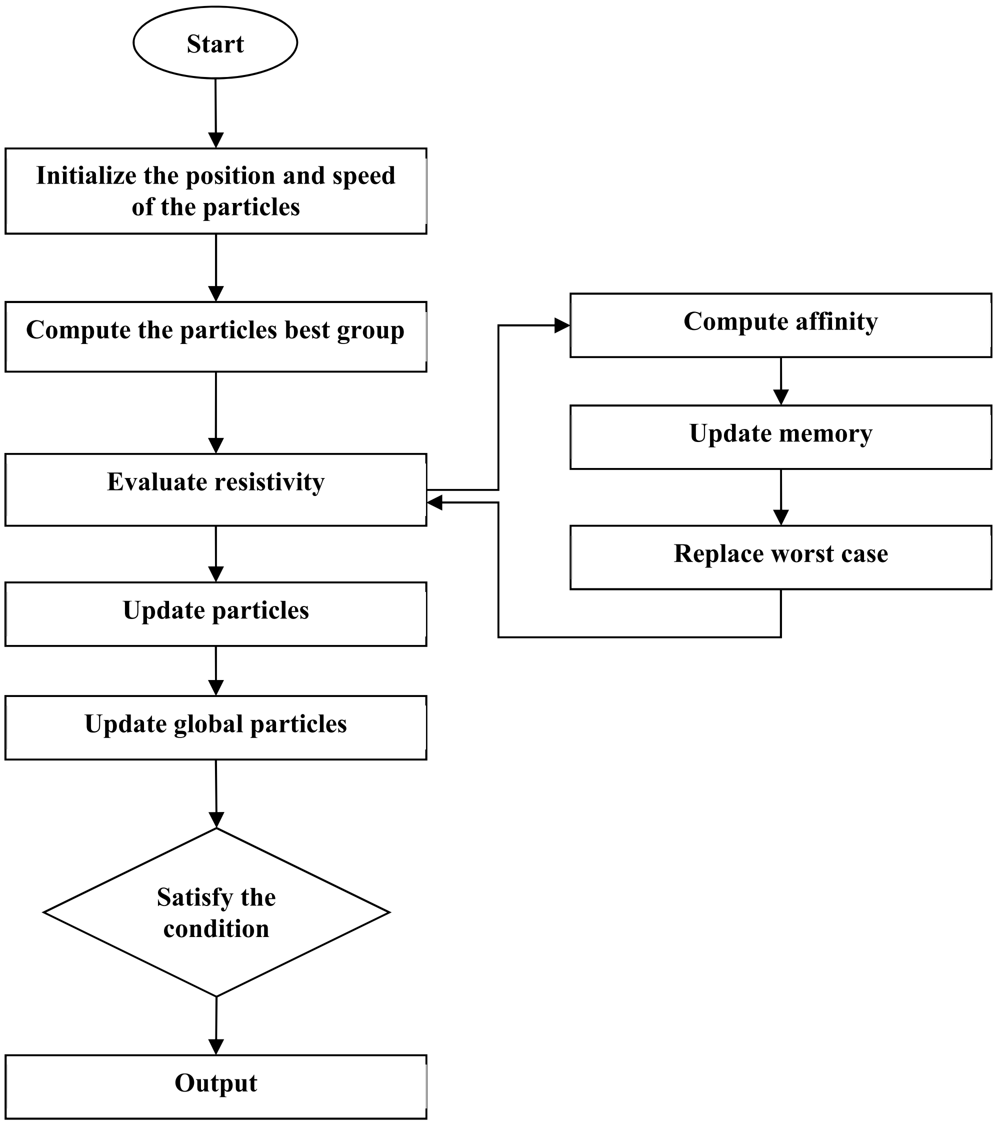

Fig. 1. Flow of resistive PSO.

The resistive particles provide scheduling solutions, and the particle dimensions are equal to the number of tasks over the flow (see Fig 1). With the provided network flow $G = \{T, E\}$ with $'N'$ tasks and the particle size is used for establishing the system coordinates for determining the search space position. Based on the infinite samples over the CC, it is highly complex to set the particle coordinates to consume arbitrary values. Therefore, the particle movement ranges based on the instance types. It is utilized to determine the candidate solution efficiency, and it is essential to reflect the objectives of the scheduling issues. The multi-objective function-based optimization problem is resolved by handling the scheduling problem where the constraint conditions and the objective functions need to be optimized. Here, the resistivity is established by candidate solutions. Thus, the affinities among the particles are provided as in Eq. (8):

$$affinity_i = (\propto cost + \beta\ makespan)/(distance_i + 1) \tag{8}$$

Here, $\propto and\ \beta$ are considered the weighted parameters and used to determine the parameters related to it. When $\propto = \beta = 0.5$, it specifies that the cost is related to makespan computation. $Distance$ is determined as the distance among the i^{th} particle to attain the global best solution, and it is evaluated using Eq. (9):

$$distance_i = \sqrt{\sum_{i=1}^{N} \left(position_{ij} - gbest_j\right)^2} \tag{9}$$

Here, $position_i$ and $gbest_i$ are the j^{th} particles and global particles. When the particle's affinity is closer to the optimal position, then the affinity is greater. It helps to compute the convergence speed with the direction of the particle move and the individuals with higher affinity for next-generation operations. The distance among the resistive particles is evaluated to measure the fall over the local optimization. It is expressed as in Eq. (10):

$$resistive\ particles_{,j} = \exp(-norm\ (ab_i - ab_j)) \tag{10}$$

When there are no resistivities towards the particles, then it causes the particles to fall under local optimization. Therefore, the non-resistive particles need to be suppressed. It is expressed as in Eq. (11):

$$resistivity = \frac{\sum_{j=1}^{N} R_{i,j}}{N} \tag{11}$$

Here, $R_{i,j} = \begin{cases} 1 & resistivity_{i,j} / \max\{resistance_{i,j}\} \geq \eta \\ 0 & resistivity_{i,j} / \max\{resistance_{i,j}\} < \eta \end{cases}$, η is constant. In some cases, immature convergence is also identified. The convergence rate is increased to eliminate local optimization and increase the population and increase individual diversity. It is expressed as in Eq. (12):

$$number_i = \frac{affinity_i}{\sum_{j=1}^{N} affinity_j} \tag{12}$$

The uniform distributions among the random numbers are shown as $[0,1]$, and the values are measured with the newly generated $' \propto '$ values. The particles are sorted based on the resistive nature, and $N - 1$ particles are chosen based on the next-generation population. Some best particles are chosen with renovated resistivity, and the worst particles are formed and replaced with a substitute. The particle speed and position are attained with optimal global solutions.

3.3 *Improved Ranking –based Grey wolf optimization* $(ir - GWO)$

Grey Wolf Optimization is a meta-heuristic that describes the nature of the wolf. The hunting nature of grey wolves captures the attention of the investigators to get the superior solution to the problem formulated. There are four levels to rank the wolves like Alpha (∝), Beta (β), Gamma (δ) and Omega(Ω). The ∝ wolf is ranked as the leader with the highest hierarchical position. β wolf ranks next to provide effective decision-making, followed by δ wolves that rank low and permit to eat less food and be dominated by others. The Ω wolves are dominative to δ wolves but dominated to ∝ and β wolves. The hunting process is assisted by ∝, β and δ, and the process include three stages like 1) Encircling, 2) hunting, and 3) attacking. The initial two stages are accountable for exploration ability while others are accountable for exploitation.

i) Encircling

Initially, the wolves are randomly positioned and defined in different search places. Thus, encircling is formulated as in Eq. (13) - Eq. (14):

$$\vec{D} = |\vec{C}.\vec{X_p}(t) - \vec{X}(t)| \tag{13}$$

$$\vec{X}(t+1) = \vec{X_p}(t) - \vec{A}.\vec{D} \tag{14}$$

Here, $'t'$ is the number of iterations, X_p is the position of prey, x is the location of the grey wolf. $\vec{A}$ and $\vec{C}$are vector coefficients, $\overrightarrow{X_p}$ is the vector of prey position, $\vec{X}$ is the position of the grey wolf. The vector coefficients $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{15}$$

$$\vec{C} = 2.\vec{r}_2 \tag{16}$$

Here, $\vec{a}$ vector has linear reduction from 2 to 0, $\vec{r}_1$ and $\vec{r}_2$ are random vectors in [0,1].

ii) Hunting

Grey wolves identify the prey's situation and surround the prey. α, β, δ are the best solutions and the others are updated based on β, δ. Generally, α, β and δ compute prey situation, and others update the solution based on the three wolves. The updation process is expressed as in:

$$\vec{X}(t+1) = (\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3})/3 \tag{17}$$

$$\begin{cases} \overrightarrow{X_1} = \overrightarrow{X_\alpha} - A_1\,(\overrightarrow{D_\alpha}) \\ \overrightarrow{X_2} = \overrightarrow{X_\beta} - A_2\,(\overrightarrow{D_\beta}) \\ \overrightarrow{X_3} = \overrightarrow{X_\delta} - A_3\,(\overrightarrow{D_\delta}) \end{cases} \tag{18}$$

$$\begin{cases} \overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}| \\ \overrightarrow{D_\beta} = |\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}| \\ \overrightarrow{D_\delta} = |\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}| \end{cases} \tag{19}$$

Where $\overrightarrow{X_1}, \overrightarrow{X_2}, \overrightarrow{X_3}$ are the three best wolves solution over the given iteration 'to. $\overrightarrow{A_1}, \overrightarrow{A_2}, \overrightarrow{A_3}$ are computed as in Eq. (18). Similarly, $\overrightarrow{C_1}, \overrightarrow{C_2}, \overrightarrow{C_3}$ are calculated using Eq. (19).

iii) Attacking

Prey hunting by wolves is accomplished when they pretend to attack. $\vec{A}$ vector reduces by reducing the$\vec{a}$ vector size to reach the prey. Subsequently, $'A'$ vector is random value between$[-a, a]$. $\vec{a}$ Value changes from 2 to 0. When the random value of $\vec{A}$ lies between $[-1, 1]$ then the successive way for searching lies between the current and the prey's situation.

Algorithm: Improved ranking-based Grey Wolf Optimization

```
Input: Initialize population of wolves, solution dimensions, range of solutions, Max_iter
Output: the best solution
Begin
Initialize a, $\vec{A}$, $\vec{C}$, and t=1
Compute fitness of each search agents of the population
$\overrightarrow{X_\alpha}$ = dominant ranking-based search agent among the population
$\overrightarrow{X_\beta}$ = dominant ranking-based search agent among the population next to $\alpha$ wolves6
$\overrightarrow{X_\delta}$ = dominant ranking-based search agent among the population next to $\beta$ wolves
While (i<Max_iter)
For each search agents
The current search agent position is updated as in Eq. (5.4)
end for
```

```
update a, $\vec{A}, \vec{C}$
Compute the fitness of all search agents
 update $\overrightarrow{X_\alpha}, \overrightarrow{X_\beta}, \overrightarrow{X_\delta}$
            i=i+1
End while
Return Best solution $\overrightarrow{X_\alpha}$
End
```

3.4 Hybridization of resistive PSO with ranking GWO

Here, random values are used for population initialization. The ranking of GWO with the initial population provides better solutions where the possibilities are provided as 0 and 1. If the random value is higher than 0.5, then the number of particles is equal to 1; else, it is zero.

i) Fitness Function

It acts as an essential role in particle searching from the search space and evaluates the degree of accuracy. The input and output values are provided with the degree of prey selection. The error prediction rate of the resistive model is described with the fitness function of $the\ hr - GWO$ approach, and a flag bit is provided to characterize the functionality (task). The solutions from the $'n'$ population, i.e. 1 are resistive particle, and 0 is non-resistive particles.

$$\text{fitness} = \varepsilon A + \varepsilon' \frac{N - LT}{N} \tag{20}$$

$$A = \frac{\text{number of available resources}}{\text{total resources}} \tag{21}$$

Where $'A'$ is the available resources, the task length is specified as LT, the total number of resistive particles are N, $\varepsilon$and $\varepsilon'$are the parameters related to the weighted vector and particle quality, ε = [0, 1] and $\varepsilon' = 1 - \varepsilon$.

ii) Population updation

Here, the velocity of every particle is evaluated with the velocity formula in continuous space and sigmoid function for mapping the multi-objective problem. It is given as in Eq. (22):

$$x_{id}^{new} = f(x) = \begin{cases} i & if(sigmoid\left(\frac{x_1 + x_2 + x_3}{3}\right) \geq rand \\ 0 & otherwise \end{cases} \tag{22}$$

Where, *and* is a random number among [0, 1]. Here, the sigmoid function is utilized for mapping continuous problem space into discontinuous problem space. The particle updation results are provided in continuous form as input and output function ranges from 0 to 1. The final value is compared with the random value and sigmoid function output results produced with a random function. When the sigmoid value is greater than the random value, then the attributes are updated; prior particles are memorized for every iteration.

Algorithm

```
Input : $NT_i, T_i, S$; c1, c2, c3, $r_1, r_2$
inertia constant: $\omega$ ;
$V_{min} = 0.1, V_{max} = 0.9$; pop_size
Output: task scheduling and resource allocation
Step 1:Begin
Step 2: Initialize Population
While (maximum iteration or convergence criteria not fulfilled)
do
While (t<max_iter)
  For i =1 to no. of particles
          Compute the fitness function
  For (d=1to number of ranking features) // * alpha, beta, gamma and evaluates the
scores and return
           best score *
  If fitness< $\alpha_{score}$
$\alpha_{score} = fitness$
$\alpha_{position} = X(i,:)$;
  end If
  If fitness $< \beta_{score}$
           $\beta_{score} = fitness$
           $\beta_{position} = X(i,:)$;
  end If
  If fitness $< \delta_{score}$
           $\delta_{score} = fitness$
           $\delta_{position} = X(i,:)$;
   end If
end For
For ( population_size)
{
     For all agents
     {
             update  $a, \overrightarrow{A, C}$
           Evaluate fitness of all search agents
            Update velocity and position based on ranking model
     end For
end For
end For
end while
Step 5: end
```

4. Numerical results and discussion

In this experimentation, the performance of the anticipated model is evaluated using MATLAB 2016b simulator. The comparison is one with various prevailing approaches like laxity bee optimization (LBP), Genetic Algorithm (GA), HEFT, Directed- Ant Colony optimization, and the proposed $hr - PSO$ with $ir - GWO$, respectively. Here, metrics like scheduling length, energy consumption (J), failure ratio (%), resource utilization (%), the average response time (Sec), and cost computation. GA performs energy consumption by eliminating task scheduling and cost computation from the models mentioned above. HEFT is depicted as the conventional scheduling model, which gives greater attention to task scheduling where the task is allocated based on priority and processing speed. Here, random graph $G = (V, E)$ is considered with the node size ranges from 20 to 100, respectively. Tasks are generally related to length and deadline, i.e. length measured with [9000, 15000] million instruction with the deadline of 5230 seconds. The processing rate and the bandwidth allocation of cloud nodes are [1000, 1500] MIPS and 2 Mbps and the $\propto, \beta, t_{max}, \varepsilon$values are set as 1, 1, 100, and 0.2, respectively. In this experimentation, the performance of the anticipated $hr - PSO$ with $their - GWO$ model is measured using the failure ratio. It is depicted as the number of tasks that are not completed within the deadline towards the total number of tasks scheduled. It is mathematically expressed as in Eq. (22):

$$Failure\ ratio = \frac{count\_failure}{count\_total} * 100\% \tag{22}$$

Based on the analysis in Table 2, it is known that the scheduling length is higher when the number of tasks increases. The scheduling length of the proposed $hr - PSO$ with $ir - GWO$ is 110, which is 35, 20, 40, and 100 lesser than other models. However, the proposed $hr - PSO$ with $ir - GWO$ gives scheduling length compared to other models. DE-ACO attains the worst-case results while the LBP shows nominal outcomes (See Fig 2). Table 3 depicts the energy comparison of the proposed model with other approaches. Here, the proposed $hr - PSO$ with $ir - GWO$ model consumes lesser energy compared to other models. LBP consumes enormous energy while the proposed model shows lesser energy. The energy consumed by the connected nodes is measured in Joules. The energy consumed by the proposed $hr - PSO$ with $ir - GWO$ is 48000J which is 4000J, 2000J, 13000J, and 10000J lesser than the prevailing approaches when 100 tasks are allocated concurrently (see Fig 3).

Table 2. Comparison of scheduling length of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Number of Tasks	**Scheduling length**				
	LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
20	110	120	130	160	100
40	140	210	280	310	120
60	90	100	120	150	75
80	130	120	130	170	100
100	145	130	150	210	110

Table 3. Comparison of energy consumption of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Number of Tasks	**Energy Comparison (J)**				
	LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
20	11000	9000	11000	10000	8000
40	22000	19000	22000	22000	17000
60	32000	30000	38000	34000	30000
80	41000	39000	48000	46000	39000
100	52000	50000	61000	58000	48000

Table 4. Comparison of Failure ratio of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Number of Tasks	**Failure ratio (%)**				
	LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
20	16%	50%	67%	31%	12%
40	22%	65%	75%	50%	19%
60	30%	60%	80%	65%	28%
80	35%	55%	82%	68%	33%
100	42%	78%	85%	80%	39%

Table 5. Comparison of Resource Utilization of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Number of Tasks	**Resource Utilization(%)**				
	LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
20	6%	15%	30%	25%	10%
40	8%	15%	20%	20%	15%
60	15%	5%	5%	15%	10%
80	5%	18%	1%	10%	15%
100	7%	16%	5%	5%	15%

Table 6. Comparison of the average response time of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Number of Tasks	**Average response time (Seconds)**				
	LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
20	7	7.5	7.8	7.9	6
40	13	16	18	20	10
60	16	18	20	23	14
80	18	20	23	24	17
100	21	23	26	27	18

Table 7. Comparison of execution cost of proposed $hr - PSO$ with $ir - GWO$ versus existing approaches.

Execution Cost				
LBP	**GA**	**HEFT**	**DE-ACO**	**hr-PSO with ir-GWO**
544500	544500	544600	543000	532000

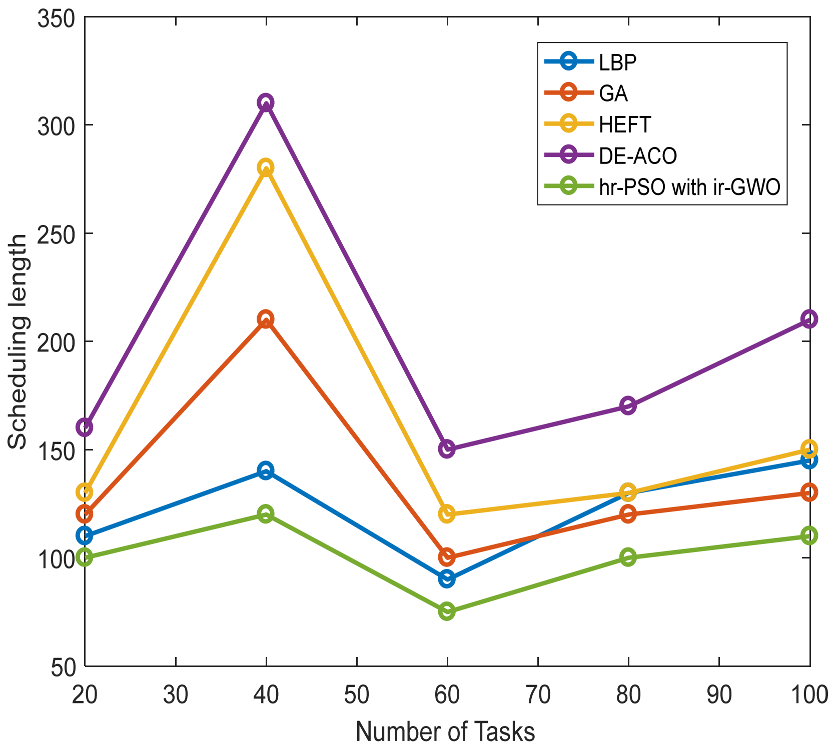

Fig. 2. Scheduling length comparison.

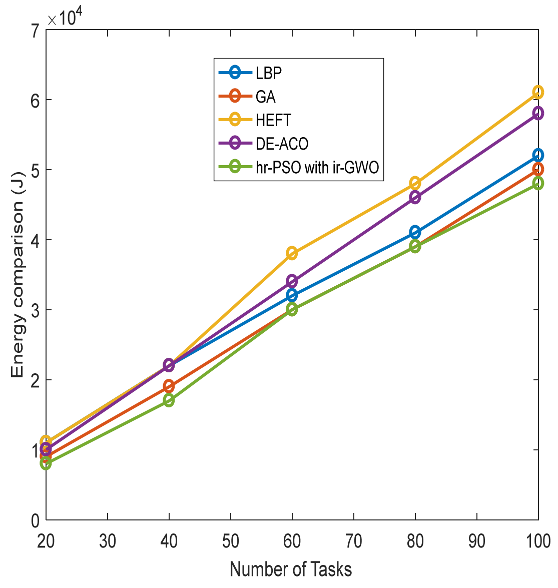

Fig. 3. Energy comparison (Joule).

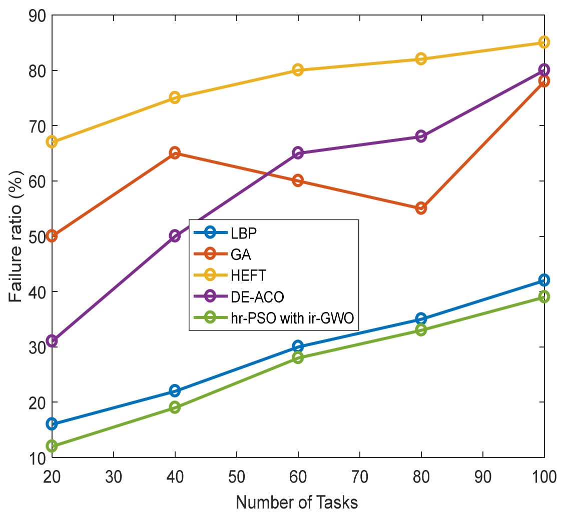

Fig. 4. Failure ratio (%) comparison.

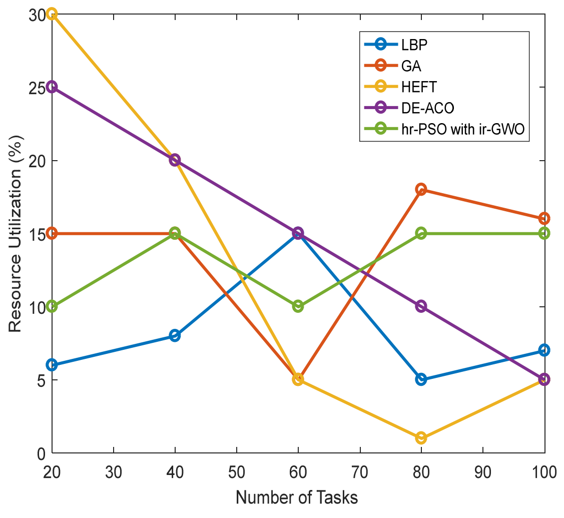

Fig. 5. Resource utilization (%) comparison.

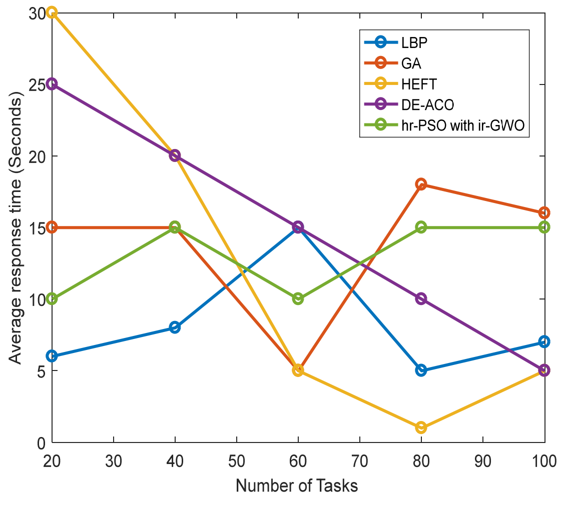

Fig. 6. Average response time (sec) comparison.

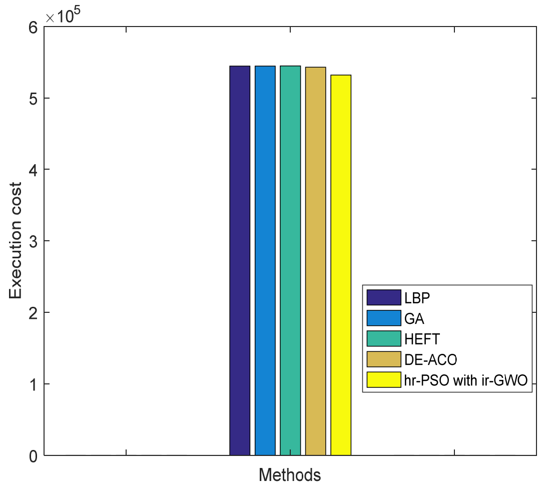

Fig. 7. Execution cost comparison.

Table 4 depicts the failure ratio comparison of proposed $hr - PSO$ with $ir - GWO$ with other approaches. The model shows a lesser failure rate than the prevailing models, i.e. 4%, 39%, 46%, and 30% lesser. It shows the significance of the model (See Fig 4). Table 5 depicts the percentage of resources utilized when 100 tasks are allocated to the model. The proposed $hr - PSO$ with $ir - GWO$ uses 15% resources while LBP uses 7%, GA uses 16%, HEFT uses 5%, and DE-ACO uses 5%. The proposed model uses nominal resources based on the required colossal energy consumption (See Fig 5). Table 6 depicts the average response time (sec) of the proposed versus existing model. The anticipated model consumes lesser time for response to the request from the end-uses while the other model takes a considerable time to respond. The proposed model takes 18 seconds which is 3 sec, 5 sec, 8 sec, and 9 sec lesser than other models (See Fig 6). Table 7 depicts the comparison of execution cost of $hr - PSO$ with $ir - GWO$ to the different approaches where the cost of proposed is 12500, 12500, 12600, and 11000 lesser than other models (See Fig 7). From the above observations, the significance of the anticipated model is projected and shows effectual task scheduling compared to LBP, GA, HEFT, DE-ACO, and so on.

5. Conclusion

This work discusses the task scheduling process over the cloud environment efficiently using a meta-heuristic model. The anticipated $hr - PSO$ with $ir - GWO$ model shows significant benefits in allocating the computing tasks at every processing node of the cloud. However, the task scheduling process is depicted as multi-objective constraints. The scheduling process relies on various metrics like energy consumption, scheduling length, failure ratio, resource utilization, average response time, and execution cost. These multi-objective constraints are significantly handled with the proposed model $hr - PSO$ with $ir - GWO$. The exploration, exploitation, and

over-fitting issues are addressed effectual by the hybridization of resistive PSO and ranking GWO. The drawbacks of the standard PSO and GWO are ensured with the hybridized model. The model attains 110 MI for 100 tasks, 48000J energy, 39% failure rate, 15% resource utilization, 18 seconds for a response, and 532000 execution cost for 100 tasks, respectively. These metrics are nominal and efficient compared to LBP, GA, HEFT, and DE-ACO models. The simulated outcomes show better and superior results than the prevailing models.
In the future, this work intends to be deployed in a real-time environment by considering various user-level analyses and executed towards different cloud nodes to resolve the incoming requests. With this plan, the initiation is taken to examine the real-world functionalities and the drawbacks of the proposed idea. However, task scheduling is considered based on associated and independent tasks.

References

[1] Deng, R., Lu, R., Lai, C., Luan, T. and Liang, H. (2016). Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption. IEEE Internet of Things Journal, pp.1-1
[2] Liu, L., Chang, Z., Guo, X., Mao, S. and Ristaniemi, T. (2018). Multi objective Optimization for Computation Offloading in Fog Computing, IEEE Internet of Things Journal, 5(1), pp.283- 294.
[3] Wu, J., Dong, M., Ota, K., Li, J., Yang, W. and Wang, M. (2019). Fog-Computing-Enabled Cognitive Network Function Virtualization for an Information-Centric Future Internet. IEEE Communications Magazine, 57(7), pp.48-54.
[4] Guan, Z., Zhang, Y., Zhu, L., Wu, L. and Yu, S. (2019). EFFECT: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. Science China Information Sciences, 62(3).
[5] Meng, X., Wang, W. and Zhang, Z. (2017). Delay-Constrained Hybrid Computation Offloading With Cloud and Fog Computing. IEEE Access, 5, pp.21355-21367.
[6] Guo, H., Liu, J. and Zhang, J. (2018). Computation Offloading for Multi-Access Mobile Edge Computing in Ultra-Dense Networks. IEEE Communications Magazine, 56(8), pp.14-19.
[7] Zhu, T., Shi, T., Li, J., Cai, Z. and Zhou, X. (2019). Task Scheduling in Deadline-Aware Mobile Edge Computing Systems. IEEE Internet of Things Journal, 6(3), pp.4854- 4866.
[8] Datta, C. B.; Haerri, J. (2015). Fog Computing Architecture to Enable Consumer Centric Internet of Things Services,''; Madrid, Spain,; pp.1-2.
[9] Zanella, N. B.; Castellani, A.; Vangelista, L.; Zorzi, M. (2014). Internet of Things for Smart Cities. IEEE Internet Things J, 1 (1), pp.22-32.
[10] Elhady; Tawfeek, M. A. (2015). A Comparative Study into Swarm Intelligence Algorithms for Dynamic Tasks Scheduling in Cloud Computing, pp 362-369.
[11] Mohammadi, R. P.; Fahringer, T. (2013) A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments. IEEE Trans. Parallel Distrib. Syst, 24 (6), pp.1203-1212.
[12] Cheng, J. L.; Wang, Y. (2015) An Energy-Saving Task Scheduling Strategy Based on Vacation Queuing Theory in Cloud Computing. Tsinghua Sci. Technol 2015, 20 (1), pp.28-39.
[13] Buyya, R. R.; Calheiros, R. N. ``Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities,''. Proc. Int. Conf. High Perform. Comput. Simulation 2009, 1 11.
[14] Jia, G. H.; Rao, H.; Shu, L. (2018). Edge Computing-Based Intelligent Manhole Cover Management System for Smart Cities,''. IEEE Internet Things J , 5 (3), pp.1648-1656.
[15] Wang et al.,(2017). QoS scheduling algorithm in cloud computing based on discrete particle swarm optimization. Comput. Eng 2017,43(6), pp.111-117.
[16] Sun, N. Z.; Wang, H.; Yin, W.; Qiu, T. (2013). PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing, Big Data, pp 243-249.
[17] Wang, G. L.; Zhao, P.; Yan, C.; Jiang, C. (2018). Behavior Consistency Computation for Workflow Nets with Unknown Correspondence. IEEE/CAA J. Autom. Sinica , 5 (1),pp. 281-291.
[18] Wang, W. Z.; Zhao, C.; Lei, Y.; Zhang, Z.; Inf. Techn Sujana,T. R.; Priya, T. S. S.; Muneeswaran, K. (2019) A Dynamic Programming-Based Approach for Cloud Instance Types Selection and Optimization. Int. J 2019, 23 (5), pp.1745-1765.
[19] Smanchat; Viriyapant, K. (2015. Taxonomies of Work_ow Scheduling Problem and Techniques in the Cloud. Future Gener. Comput. Syst , 52, pp.1 -12,.
[20] Sujana, T. R.; Priya, T. S. S.; Muneeswaran, K. ``Smart PSO- Based Secured Scheduling Approaches for Scienti_c Work_ows in Cloud Computing,''. Soft Comput 2019, 23 (5),1745 1765.
[21] Mohan, M. E.; Lu, S.; Kotov, A. (2016) ``Scheduling Big Data Workflows in the Cloud under Budget Constraints, pp.2775-2784.
[22] Jiang, H. E.; Song, M. (2017). Dynamic Scheduling of Workflow for Makespan and Robustness Improvement in the IaaS Cloud,''. IEICE Trans. Inf. Syst, D (4), pp. 813-821.
[23] Juve, A. C.; Deelman, E.; Bharathi, S.; Mehta, G.; Vahi, K.(2013). characterizing and profiling scientific workflows. Future Generation. Computer System, 29 (3), pp. 682-692.
[24] Silva, W. C.; Juve, G.; Vahi, K.; Deelman, E. (2014). Community Resources for Enabling Research in Distributed Scientific Workflows, p 177-184.
[25] Shah-Mansouri, V. W. W.; Schober, R. (2017). Joint Optimal Pricing and Task Scheduling in Mobile Cloud Computing Systems. IEEE Trans. Wireless Communication. 16 (8), pp.5218-5232.

Authors Profile

Rajkumar Kalimuthu Pursing Ph.D. in Computer Science Engineering, Noorul Islam Centre for Higher Education, India. He has completed his Master of Engineering in Software Engineering from Mount Zion College of Engineering and Technology, Anna University, India and he has Completed B.E (Computer Science and Engineering) degree in Srinivasan Engineering College, Anna University, India. He has interest in the field of cloud computing, Data Mining, Software Engineering and Networking

Dr. Brindha Thomas is an Associate Professor in Noorul Islam Centre for Higher Education, India. She received Ph.D degree in Cloud Computing from Noorul Islam University, India. She has completed her ME degree in Software Engineering from Periyar Manniammai College of Engineering, Anna University, India and her B.Tech degree from Jayamatha Engineering College, Anna University, India. Her area of interest is cloud computing, service oriented computing, parallel processing and distributed computing.