Received: December 29, 2022. Revised: April 5, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Task Scheduling Based on Cost and Execution Time Using Ameliorate Grey Wolf Optimizer Algorithm in Cloud Computing

Manikandan Nanjappan[1*] Pradeep Krishnadoss[2] Javid Ali[3] Gobalakrishnan Natesan[4] Balasundaram Ananthakrishnan[2]

[1]*SRM Institute of Science and Technology, Kattankulathur, Chennai, TN, India*
[2]*Vellore Institute of Technology, Chennai, Tamilnadu, India*
[3] *St.Joseph's Institute of Technology, Chennai, Tamilnadu, India*
[4] *Sri Venkateswara College of Engineering, Chennai 602117, TN, India*
*Corresponding author's Email: macs2005ciet@gmail.com

Abstract: An enriched computational platform has been unfolded with the introduction of cloud technology offering ensemble services to users that includes storage, database and processing power. More recently, the cloud technology has been upgraded to a federated environment that offers even more features where various service providers could interconnect for providing an integrated service in a transparent way to cloud users. Applications that demand enormous computing resources like for instance bioinformatics workflow applications could very well make use of the abundant cloud resources for effective execution. Fine tuning the task scheduling activity in cloud could further boost the overall cloud performance. In this paper, we had designed an optimal and ideal task scheduling algorithm that primarily focuses on reducing the cost and makespan QoS parameters, eventually leading to enhanced cloud performance. The proposed algorithm, which is a meta-heuristic enhanced hybrid version named, grey wolf optimizer cuckoo (GWOC) is formally designed from the existing grey wolf optimizer and cuckoo search algorithms. results obtained clearly justify the goal accomplishment of the proposed GWOC algorithm and its swiftness in achieving convergence, thereby clearly outperforming existing contemporaries like gravitational search algorithm (GSA), whale optimization algorithm (WOA) algorithm and grey wolf optimizer (GWO) algorithm. The proposed GWOC technique had produced an improvement of 2.11%, 3.5% and 5.17% for makespan and had reduced the cost to the tune of 7.71%, 11.3%, and 15.4% when compared with gravitational search algorithm (GSA), whale optimization algorithm (WOA) algorithm and grey wolf optimizer (GWO) algorithm respectively when used with 100 VMs. Detailed results have been presented in section 5.

Keywords: Cloud computing, Task scheduling, Cost, Execution time, Cuckoo search, Grey wolf optimizer.

1. Introduction

Multi-platform applications like mobile computing, networking, environmental, medical, business rely heavily on cloud technology [1], for the processing and storage services. The cloud technology is the most sought after when it comes to providing IT related services. Companies looking to invest on huge infrastructural components could be highly relieved by making utmost advantage of the pay-per-use model of cloud environment. Cloud services could be broadly categorized under three sections namely software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). Virtual resources based on demand could be offered to users for completion of their respective tasks. Flexible resources from cloud repository are engaged for effective service accomplishment. As users need to pay for actual services rendered from cloud, the term pay-per-use became popular. Optimizing the scheduling activity in cloud enormously increases its performance, thereby attracting more users towards it.

The task scheduling activity in cloud is of NP-hard type. In IaaS type of service, server clusters are engaged for providing the required service where in addition to hardware resources, software resources too need to be efficiently managed in an elastic manner for effective task completion. Faults occurring in these resources too need to be overcome in a transparent manner such that task execution proceeds unhindered. Scheduled tasks are taken care off by the resource management subsystem present in the IaaS cloud that maps the tasks in an efficient manner to available VMs that exhibit the dynamic and heterogeneous characteristics. Heuristic approaches yield better optimal solutions in these type of NP-problem cases [2-4].

Reducing the cost and execution time add up to cloud performance. The VMs in cloud are equipped with heterogeneous processing capability offering load balancing feature among virtual machines, thus offering better co-ordination and efficient task scheduling, thereby realizing the reduced makespan requirement. The task scheduling algorithm needs to efficiently handle the execution time of tasks and load balancing among the VMs. In this paper, a hybridized multi-objective task scheduling algorithm called grey wolf optimizer cuckoo (GWOC) had been proposed. GWOC had been designed by integrating the grey wolf optimization algorithm and cuckoo search algorithm for effectively managing the makespan and cost parameters. The multi-objective function designed, helps in achieving a near optimal solution to the scheduling activity in cloud. The major contribution made in this work pertaining to task scheduling includes:

1. Task scheduling is carried out using an innovative approach namely grey wolf optimizer Cuckoo (GWOC) by integrating the grey wolf algorithm and Cuckoo search algorithm for obtaining a near optimal solution.
2. The original GWO algorithm usually gets confined to local optima while carrying out exploration and exploitation activities. This shortfall had been rectified by integrating it with Cuckoo Search algorithm. Also, the Cuckoo search is inherently slow and has low convergence. These features of Cuckoo search algorithm get balanced by integrating it with GWO algorithm which comparatively scores well in these areas.
3. Makespan and cost are the QoS parameters optimized in this work whose effective management during task scheduling leads to enhanced cloud performance.
4. Experiments had been conducted in a simulated environment using CloudSim 3.0 toolkit.

This paper herewith had been organized as follows: section 2 contains an overview of recent literatures pertaining to task scheduling in cloud environment. The solution framework and problem description are detailed out in section 3. Section 4 describes the proposed hybrid multi-objective Grey Wolf Optimizer Cuckoo (GWOC) task scheduling algorithm. Section 5 presents the experimental evaluations and discussions based on the results obtained and finally, conclusion notes are presented in section 6.

2. Related work

The authors [5] had analyzed the huge energy consumption during task scheduling where a received task is mapped on to a suitable virtual machine in cloud and had attributed it to absence of efficient task scheduling algorithms that could provide a solution to this issue. The authors had presented an approach based on energy and cost-aware scheduling (ECWS) named as heterogeneous earliest finish time (HEFT) based energy-efficient heuristic for cloud scheduling. The authors had validated the efficiency of their algorithm by conducting simulated experiments using WorkflowSim tool with respect to energy conservation, resource utilization and cost parameters. Their proposed algorithm showed promising results when compared to other existing algorithms.

The authors [6] had pondered upon the secure allocation of virtual machines in an energy efficient manner inside a cloud data center so that wastages in the form of resource, power consumption, cost and security flaws could be efficiently contained. The authors had designed an innovative, secure and multi-objective VMP (SM-VMP) framework to address these issues. The authors had proposed the whale optimization genetic algorithm (WOGA) for efficient placement of virtual machines such that the VMs could be easily allocated resulting in reduced delay, cost and addressing security concerns as well. The experimental validation had revealed that the algorithm proposed had yielded commendable results when compared to other existing algorithms.

The authors [7] had proposed the multi-objective energy aware genetic algorithm whose primary objective is to reduce the energy consumed and execution time that could result in optimal scheduling of cloud tasks. Energy consumed by the CPU in the virtual machines had been considerably reduced. Tasks, based on a fitness function are chosen based on an energy model. The authors had validated the performance of their proposed algorithm by comparing it with other existing algorithms with respect to cost and energy parameters. The obtained results establish the superiority of the proposed algorithm.

The authors [8] had addressed the issue of efficiently allocating cloud services to various competing tasks such that execution cost, makespan are optimized in accordance to user specified constraints. The authors had developed a framework named autonomic resource provisioning and scheduling (ARPS) framework that could independently make decisions pertaining to scheduling of tasks on best available resources such that execution time and cost gets optimized as per user defined constraints. Scheduling inside the framework had been carried out using the spider monkey optimization (SMO) algorithm. The proposed algorithm effectively reduced the cost, time and energy consumption parameters when compared to other existing algorithms.

Fog computing had been effectively placed between internet of things (IoT) users and the cloud layer for providing solutions to real time applications. For Internet of Things application balancing the load among available resources turns out to be more crucial to achieve optimal solution. The authors [9] had proposed fog computing architecture of load balancing (FOCALB) suitable to scientific workflow type of applications. A hybrid algorithm for balancing the load in workflow type of applications had been designed by the authors by integrating the tabu search, grey wolf optimization (GWO), and ant colony optimization (ACO) algorithms. The proposed algorithm increases the resource utilization at the fog layer by effectively balancing the load. The authors had compared the performance of their FOCALB model with other existing algorithms based on execution time, cost, and energy consumption parameters. Results obtained justify the efficiency of the proposed model.

The authors [10] had proposed a novel hybridized algorithm named chemical reaction partial swarm optimization for efficient task scheduling in cloud. The independently received multiple tasks had been allocated to available virtual machines. The authors had performed hybridization by integrating the optical schedule sequence so that the processing of tasks adheres to the demand and deadline constraints. The authors had focussed on the betterment of QoS parameters like makespan, cost and energy consumption.

By maximizing the throughput and minimizing the response time, consuming less energy and utilizing the resources to the maximum level the cloud performance can be enhanced. Optimization of task scheduling in cloud accounts for maximum throughput, minimum response time, reduced energy consumption and optimal utilization of resources. The authors [11] had proposed a Hybrid ant genetic algorithm for efficiently scheduling the tasks in cloud. The authors had divided the tasks and virtual machines into small clusters by combining the features of genetic and ant colony algorithms. After task are allocated, pheromone is supplemented to the virtual machines. Loaded virtual machines are effectively identified by the proposed algorithm leading to reduced solution space. This results in faster convergence and reduced response time. He proposed algorithm accounted for minimizing the running time workflow tasks.

High performance applications (HPC) can be effectively executed on cloud platforms with the high-level resources present there. However, higher exploitation of cloud resources for such type of applications results in increased energy consumption, increased cost and higher carbon emission in environment. In addition to this, users demand on want of QoS services like cost, execution time, resource utilization further increase the load in the cloud computing environment. To address these demands, the cloud scheduling activity needs to be overhauled or needs to be optimized further. The authors [12] had introduced a QoS-aware energy-efficient scheduling policy named CSPSO, for efficient scheduling of tasks in cloud to optimize the energy consumption and minimize the makespan of given tasks. The proposed CSPSO is a hybridized version of Cuckoo search and particle swarm optimization algorithms, achieving quick convergence rate and overcomes the absence of diversity standard of Cuckoo search algorithm. Resources were allocated efficiently through a fitness-aware resource allocation (FARA) strategy. Also, the authors had incorporated a velocity update mechanism for cuckoo individuals in their proposed CSPSO methodology.

Achieving optimality among the available cloud resources is always challenging. Fine tuning the scheduling activity in cloud helps in fixing this issue to a great extent. The authors [13] had proposed innovative scheduling technique based on SLnO and multi-objective model to address this issue. The main objective of the proposed algorithm is to minimize the execution time, cost and energy consumption by optimally making use of available resources in cloud environment. Experimental results and the subsequent comparative analysis conducted between the proposed algorithm and other contemporary approaches like vocalization of whale optimization algorithm (VWOA), whale optimization algorithm (WOA), grey wolf optimization (GWO) and round robin (RR) algorithms had shown that the proposed approach fares better with respect to these QoS parameters.

The authors [14] had proposed a multi-objective optimization algorithm named modified fractional grey wolf optimizer for multi-objective task scheduling (MFGMTS) for optimizing the cloud performance by fine tuning the scheduling process. The authors had considered a host of QoS parameters like which they had assessed using the epsilon-constraint and penalty cost function criteria. The proposed algorithm had been fine tuned in accordance to the fractional grey-wolf optimization (FGWO), where in this case, the positions are updated differently by including an additional term by applying a combination of alpha and beta solutions. The performance efficiency of MFGMTS had been assessed by comparing it with the present particle swarm optimization, genetic algorithm (GA), grey wolf optimizer, and FGWO algorithms.

The scheduling activity involving scientific tasks is of NP hard problem type. Recent solutions for optimizing the scheduling activity had been proposed by many researchers. Improper scheduling of tasks pulls down the cloud performance. Service quality and energy consumption are vital for realizing an enhanced cloud performance. The authors [15] had designed an improvised task scheduling algorithm by integrating the Cuckoo search algorithm (CSA) and whale optimization algorithm (WOA), named the CWOA to enhance the cloud performance.

A variety of multi-objective works had been carried out in various domains to achieve performance excellence. Many authors had applied the optimization features in electrical, health care domains and even for environmental assessments like identifying groundwater quality. A slew of researchers had incorporated the preying and mating features of animals and had mapped their behaviours in attaining optimal solutions adopting multi-heuristic approaches [16-24].

Hence in spite of considerable research, there seems to be a dearth of improvisation to makespan and cost parameters when considered together. Therefore, the GWOC algorithm had been proposed for making considerable improvements to the makespan and cost parameters. In GWOC, the conventional grey wolf optimization algorithm that falls prey to local optima and Cuckoo search algorithm that has low convergence accuracy had been integrated to improvise the task scheduling activity in cloud. The disadvantage of each of these algorithms gets balanced by the other upon integration. The proposed GWOC had been validated for its performance efficiency by comparing it with GSA, WOA and GWO algorithms and the detailed results have been portrayed in the results section.

3. Solution framework

The proposed GWOC focuses on minimizing the makespan and cost during the task scheduling for enhancing the cloud performance. By assigning the submitted tasks to appropriate resources that are currently available, the scheduling could be optimized, leading to improved cloud performance. Users, submitting their tasks to cloud know priorly about the type of tasks they are submitting for competent servicing. The Task Repository creates an aggregation of such user submitted tasks. Tasks are submitted for servicing from a task queue. The demands to be met in the form of QoS constraints are revamped into an objective function that has to be satisfied with appropriate data for meeting the proposed objective. The tasks queued up in Task queue result in the formation of a task list (TL) from where they are presented to the cloud environment. Tasks from task queue are studied by the task analyzer (TA) and eventually re-organizes their servicing order based on priority of each individual task and the same has been depicted in the below figure in the form of a re-organized queue, where each task is re-arranged on the basis of its completion time. These tasks are taken up for servicing by the Task Scheduler which it assigns to appropriate cloud resources for providing the requested service. Several criteria including place of data, task limitations, load on available resources are carefully considered by the task scheduler during this operation.

The scheduling activity is carried out based on the necessary information provided by the task manager and the resource manager (RM). The TIP has been assigned with the responsibility of providing data with respect to the location and copies of several varieties of the application data. The resource manager has been assigned the responsibility of tracking the currently available resources that could be assigned to incoming tasks. Whenever, there is an imbalance between the tasks and available resources, the resource manager looks in to other cloud providers that are capable of offering the required resources in a more economical manner and assigns the additional tasks to them. Once an appropriate resource is identified, the task manager (TM) assigns the task to it.

The TM also monitors the progress of each of such assigned task throughout its completion life cycle. The framework of the proposed work had been depicted in Fig. 1.

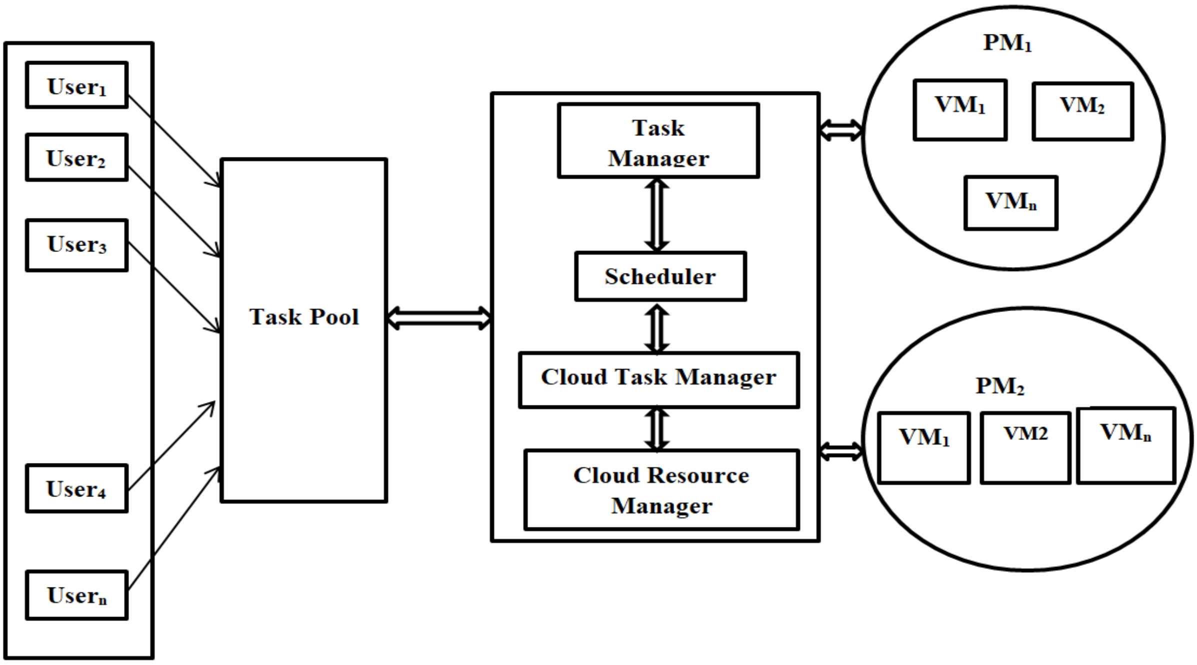

Figure. 1 Architecture diagram of GWOC

4. Problem definition with solution framework

The GWOC had been proposed to minimize the makespan and cost expended during the task scheduling activity that could result in increased cloud performance. In this proposed work, scheduling of tasks happens in parallel where each task is processed simultaneously. The scheduling activity analyses the resources that may be needed to complete any given task and which resource is capable of completing the given task.

Before parallelism kicks off, larger tasks could be fragmented into smaller sub-tasks. This results in increasing the gain of entire computation since now each of the sub-task could be run on more than one processor. Allocating the received tasks on identified resources by maintaining the priority constraints is the prime goal of any task scheduling algorithm and indeed, is daunting.

Hence the GWOC approach has been proposed in this work for minimizing the makespan and cost for improving the cloud performance.

Each task that had been submitted comprise of a number of simultaneous as well as autonomous jobs. All of these jobs had to be executed on a single identified instance. Let $PM = \{PM_1, PM_2, ..., PM_n\}$ denote the cloud physical machine set, $VM_i = \{VM_1, VM_2, ..., VM_I\}$ denote the virtual machine set and $T = \{T_1, T_2, ..., T_m\}$ denote the task set to be executed.

Table 1. Notation used in GWOA algorithm

Symbol	Descriptions
VMs	Virtual-Machines
PMs	Physical-Machines
T_{ij}	Tasks j, 1< i < K
C_i	Cost
E_i	Execution Time
α, β	Parameters used for controlling
CPU_{ij}	Total CPU Capacity.
PC	Capacity of the Processor
TL	Length of one particular Task

The objective function of the proposed work could be stated as:

$$Objective\ function = \sum_{i=1}^{m} T_i \; (\alpha \cdot cost + \beta \cdot Executiontimes) \quad (1)$$

The QoS parameters taken into account in the proposed GWOC are cost and execution time. They are described below:

Cost: The cost incurred in executing a task in an optimal manner could be deduced by calculating the ratio between the number of virtual machines that are moved up or allotted with that of total number of virtual machines that are actually available on any given physical resource. The cost parameter could be deduced as shown in equation given below:

$$C_i = \frac{1}{PM} \sum_{i=1}^{m} \left(\frac{No.of\ movement\ in\ VM}{Total\ VMs} \right) \quad (2)$$

Where C_i indicates the measure of cost and PM indicates the physical machine.

Execution time: The execution time denotes the time taken by the proposed algorithm to complete a task that is expressed as the ratio between the task's length with that of the processing capacity of the VM. The execution time parameter could be deduced as shown in equation given below:

$$E_i = Task^{Length} / Processor^{Cabacity} \tag{3}$$

Where E_i denotes the execution time, *TL* indicates the task's length and *PC* denotes the processor's capacity

4.1 Proposed GWOC based scheduling approach

The overall architecture of the proposed approach had been shown in Fig. 1 above. The tasks that are submitted by the user are forwarded to the Task Scheduler. The Task is assigned by the Task Scheduler in accordance to the fitness function measure, computed for each task. The Resource Manager tracks the availability of virtual machines and other resources. The proposed GWOC takes up the goal of minimizing the cost and execution time of each scheduled task.

4.2 Proposed algorithm of GWOC

Grey wolf optimization algorithm

4.2.1. Overview

The grey wolves could be categorized into four types namely alpha (α), beta (β), delta (δ) and omega (ω). These four groups usually live together. The category of Alpha (α) regarded as a pioneer wolf type is located at the highest point of the pyramid. Though it has strong administrative capabilities, it may not be strong enough and usually articulates significant decisions to their gathering, like for instance the predation sequence or decision with respect to food circulation. Next in the order are the beta (β) types that provide assistance to Alpha (α) group in managing other groups. Its prime responsibility is to obey the 'α' group and simultaneously issue commands to other sub-groups. Next in the pyramid are the delta (δ), that carry out command instructions from both α and β groups. Once the 'α' and 'β' groups grew older, they are adjudicated with the decision-making responsibility. The omega (ω) group make up the last section of the pyramid and their responsibility is to carry out the instructions issued by other higher groups.

4.2.2. Mathematical model

In the proposed work, the arrangement of grey wolves is decided by their fitness capability. It could be seen from the proposed fitness equation that the superlative solution is offered by the α, β, and δ wolves. These three wolf categories had been regarded as key types in the proposed work and the remaining category is the omega – ω type. The predation procedure [25] had been parted into three stages as described below:

4.2.3. Encircling prey

$$\vec{W} = \left| \vec{X} \,.\, \overrightarrow{N_p}(t) - \vec{P}(t) \right| \tag{4}$$

$$\vec{Y}(t+1) = \overrightarrow{P_p} * (t) - \vec{Z} \,.\, \vec{W} \tag{5}$$

Where (t+1) denotes the subsequent iteration, $\vec{W}$ denotes wolf location, $\overrightarrow{P_p}$ indicates the prey location, $\vec{Z}\ and\ \vec{W}$ are the coefficient vectors. The estimation process had been depicted in the below equations:

$$\vec{Z} = 2\vec{e} \,.\, \overrightarrow{n_1} - \vec{e} \tag{6}$$

$$\vec{W} = 2.\overrightarrow{n_2} \tag{7}$$

Where $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$ point to a number in between 0 – 1, $\vec{e}$ vector denotes iterations in the range between 2 and 0.

4.2.4. Hunting prey

Once prey location had been identified, the α, β and δ wolves direct other wolf types to start encircling the prey and assist to zeroing on the best solution, subsequently updating the current position of each, based on the key gathering. Current location update had been done based on the equation given below:

$$\vec{Y}(t+1) = \frac{\overrightarrow{Y1} + \overrightarrow{Y2} + \overrightarrow{Y}3}{3} \tag{8}$$

$$\overrightarrow{P^1} = \left| \overrightarrow{P_\alpha} - \overrightarrow{C^1}.\overrightarrow{A_\alpha}) \right| \tag{9}$$

$$\overrightarrow{P^2} = \left| \overrightarrow{P_\beta} - \overrightarrow{C^2}.\overrightarrow{A_\beta}) \right| \tag{10}$$

$$\overrightarrow{P^3} = \left| \overrightarrow{P_\delta} - \overrightarrow{C^3}.\overrightarrow{A_\delta}) \right| \tag{11}$$

Where $\overrightarrow{P_\alpha}$, $\overrightarrow{P_\beta}$ and $\overrightarrow{P_\delta}$ denote the best solution identified till now from the iterations carried out. A host of parameter measures had been deduced from the following equations:

$$\overrightarrow{A_\alpha} = \left|\overrightarrow{B_1} - \overrightarrow{N^\alpha}.\overrightarrow{A_\alpha})\right| \tag{12}$$

$$\overrightarrow{A_\beta} = \left|\overrightarrow{B_2} - \overrightarrow{N^\beta}.\overrightarrow{A_\beta})\right| \tag{13}$$

$$\overrightarrow{A_\delta} = \left|\overrightarrow{B_3} - \overrightarrow{N^\delta}.\overrightarrow{A_\delta})\right| \tag{14}$$

4.2.5. Attacking prey

Usually, the grey wolves attack their prey once it becomes stationary. Therefore, the movement of the grey wolves towards the prey could be ascertained by the following equation:

$$A = 2 - 2\left(\frac{T_S}{T_{Max}}\right) \tag{15}$$

Where T_S indicates the current iteration count t, a whole number and T_{Max} indicates the maximum number of cycles.

4.3 Modified grey wolf optimization algorithm

Population algorithms deal with two phases namely the exploration phase and exploitation phase to deduce a global best point. The changeover from exploration to exploitation phases in the GWO algorithm had been expressed in terms of the adaptable α and A estimations. When the measure of $|A| \geq 1$, the exploration phase is under progress and whenever the value of $|A| < 1$, exploitation phase is taken up. It could be seen from Eq. (10) that in the basic GWO algorithm the measure of A decreases straight from 2 to 0. In the case of the proposed GWOC, higher number of iterations are allocated to the exploration phase and only fewer to the exploitation phase. A's value measure gets reduced from 2 to 0 in accordance to shown below:

$$A = 2 - 2\left(\frac{T^{2.5}{}_S}{T^{2.5}{}_{Max}}\right) \tag{16}$$

In GWO algorithm, the positions of top three categories of wolves are deduced as depicted in Eq. (5). In the proposed GWOC, priority is considered based on which a measure of weight is assigned to these three top categories of wolves like for instance, the class of α are weighed more compared to classes of β and δ wolves. Assigning priority helps in further refinement of ideal solution as compared to the original GWO technique. This is described in the equation below:

$$\vec{Y}(t+1) = \frac{3\overrightarrow{Y_1} + 2\overrightarrow{Y_2} + \overrightarrow{Y_3}}{6} \tag{17}$$

4.4 Cuckoo search algorithm

The Cuckoo Search algorithm had been inspired from nesting habits of the cuckoo bird and the levy-flight elegance. A few cuckoos have special reproduction strategy. These cuckoo birds identify other bird's nest that are currently unoccupied and lay their eggs in them. In a few circumstances, the host birds spot the presence of outside bird's eggs in their nest and just drop them off from their nest. Levy flight is the standard flight practice of a range of creatures. It generally starts up with little steps for most of the part end eventually gathers momentum to a long-range jump. Also, there is a high probability of notable deviation from the usual mean estimation process akin to the feature of CS algorithm that bounces off from the local optimum [26]. The standard working of CS algorithm could be summarized in three rule points as given below:

1. The cuckoo bird randomly spots a home nest with just one egg.
2. The nest condition determines whether it could be adopted for future laying of eggs.
3. Two things remain as such, one, the number of birds' nest and second, the possibility of an egg getting spotted.

By adhering to above mentioned rules, revitalization of the nests is done based on the below conditions during the iteration trial.

$$\overrightarrow{S_j}(s+1) = S_j(s) + \omega \oplus Levy(\gamma), j = 1,2,3 \ldots n) \tag{18}$$

where $\oplus$ denotes entry-wise multiplication, $\overrightarrow{S_j}(s+1)$ indicates the newer solution for cuckoo j, S_j indicates the present best solution, and ω = 1 denotes the progression size. Then, the Levy-weight can be assessed as shown in Eq. (19) given below:

$$Levy \sim \mathrm{u} = t^{-\gamma}, (1 < \gamma < 3) \tag{19}$$

4.5 Algorithm of GWOC

Input:

A set of tasks, a set of host machines, a set of virtual machines

Output:

Minimized makespan and cost

GWOC Algorithm

1. Let max_iter indicates maximum number of iterations and agent indicates wolf.
2. Let Xi (i=1, 2..........) indicates the initial population of grey wolves.
3. Initialization of parameters e, A and C
4. Begin evaluating the fitness for every search agent
5. Let X α be denoted as the best search agent
6. Let X β be denoted as the second-best search agent
7. Let X δ be denoted as the third-best search agent
8. While (t remains lesser than max_iter) do
   for every search agent do
   revamp the current search agent location by following Eqs. (8-11)
   end for
   e, A and C are updated
   Assess fitness of all search agents
   X α, X β and X δ are updated based on Eq. (17)
   Further, the position of X α, X β and X δ is updated based on Eqs. (18) and (19)
   Increment the value of t
   end while
9. return X α

5. Results and discussion

Experimental evaluation of the proposed GWOC task scheduling approach had been carried out using Cloudsim simulation toolkit. Experiments had been carried out on a PC configured with Windows 7 operating system, 2 GHz dual core with 4 GB main memory processor and running a 64-bit version of Windows 2007 and JDK 1.6 platform. The parameters that had been taken up for performance evaluation are makespan and cost. The input task had been varied from 100 to 1000 numbers. 100 Vms and 200 VMs were considered for execution purpose. The proposed GWOC had been compared with GSA [27], WOA [15] and GWO [28] algorithms.

5.1 Comparison of makespan

Task execution time is vital in deciding the performance of any task scheduling algorithm and the entire cloud as well. The proposed GWOC approach has been compared with GSA, WOA and GWO algorithms based on the makespan parameter. In the below Fig. 2, results obtained when considering 100 VMs had been shown for these approaches. It could be seen that for a task count of 250, the result obtained are 178.35,187.4,191.12 and 192.2 for GWOC, GSA, WOA and GWO algorithms respectively. Upon increasing the task numbers to 500, the results obtained are 290.2, 294.1, 302.12 and 310.4 for GWOC, GSA, WOA and GWO algorithms respectively.

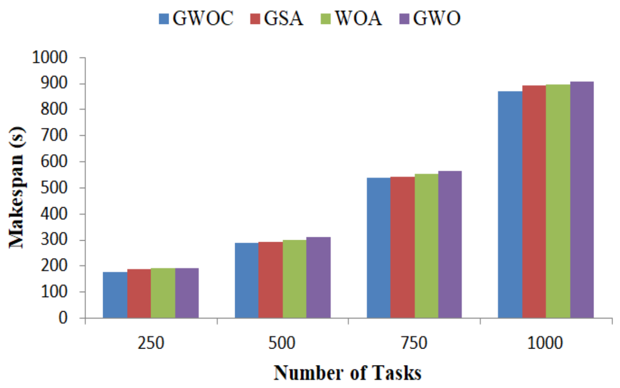

Figure. 2 Makespan of 100 VMs

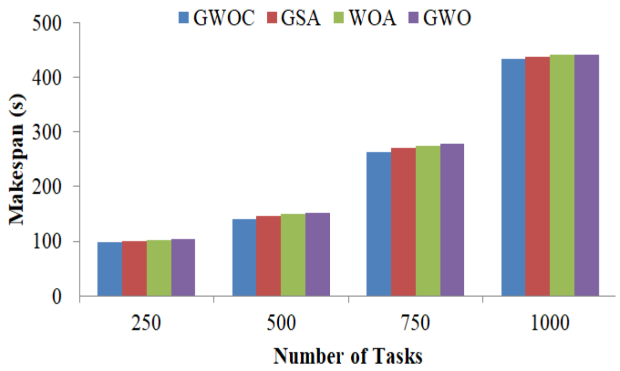

Figure. 3 Makespan of 200 VMs

Similarly for 750 tasks, the makespan values obtained are 538.4, 543.4, 554.7 and 566.23 for GWOC, GSA, WOA and GWO algorithms respectively. Finally for 1000 tasks, the GWOC, GSA, WOA and GWO algorithms had produced values of 871.3, 893.1, 897.23 and 906.6 respectively.

In Fig. 3, 200 VMs were considered for deducing the performance of the proposed GWOC. The obtained values had been compared with GSA, WOA and GWO algorithms with respect to the makespan parameter. Here too tasks are varied from 100 to 1000 numbers. It could be seen that for a task count of 250, the result obtained are 97.34, 100.01, 102.01 and 104.13for GWOC, GSA, WOA and GWO algorithms respectively. Upon increasing the task numbers to 500, the results obtained are 140.5, 146.6, 149.7 and 151.67 for GWOC, GSA, WOA and GWO algorithms respectively. Similarly for 750 tasks, the values obtained are 263.5, 271.5, 274.5 and 278.8 for GWOC, GSA, WOA and GWO algorithms respectively. Finally for 1000 tasks, the GWOC, CS, PSO and GWO algorithms had produced values of 434, 437.6, 441.6 and 442.1 respectively. Hence, it is clear that the proposed GWOC outwits the GSA, WOA and GWO algorithms with respect to the makespan parameter. The makespan results obtained using the proposed GWOC are far better when compared to the other algorithms considered. Also, this value difference increases gradually for task count of 250, 500, 750 and 1000 numbers signifying that GWOC yields improved performance with respect to the makespan parameter.

Hence it can be observed that the proposed GWOC technique, when used with 100 VMs, had produced an improvement of 2.11%, 3.5% and 5.17% for makespan, when compared with GSA, WOA and GWO algorithms respectively. Also it had produced an improvement of 2.17%, 3.4% and 2.2% for makespan when compared with GSA, WOA algorithm and GWO algorithm respectively when used with 200 VMs.

5.2 Comparison of cost

The proposed GWOC approach had been evaluated for its performance by comparing it with GSA, WOA and GWO algorithms with respect to the cost parameter. Results obtained for 250, 500, 750 and 1000 numbers of tasks with respect to cost had been evaluated. The plots are obtained for 100 and 200 VMs. In Fig. 4, 100 VMs are considered. It could be seen that for a task count of 250, the result obtained are 126.73, 133.02, 142.53 and 159 for GWOC, GSA, WOA and GWO algorithms respectively. Upon increasing the task numbers to 500, the results obtained are 202.4, 231.12, 248.03 and 260.2 for GWOC, GSA, WOA and GWO algorithms respectively. Similarly for 750 tasks, the values obtained are 362.45, 382.5, 388.5 and 394.2 for GWOC, GSA, WOA and GWO algorithms respectively. Finally for 1000 tasks, the GWOC, GSA, WOA and GWO algorithms had produced values of 541.3, 581.4, 593.7 and 610.23 respectively. In all these cases, it could be seen that the proposed GWOC approach is clearly ahead of the other algorithms with respect to the cost parameter.

In Fig. 5, 200 VMs are considered. It could be seen that for a task count of 250, the result obtained are 69.2, 82.23, 87.81 and 98.8 for GWOC, GSA, WOA and GWO algorithms respectively. Upon increasing the task numbers to 500, the results obtained are 111.3, 121.7, 132.21 and 140.4 for GWOC, GSA, WOA and GWO algorithms respectively. Similarly for 750 tasks, the values obtained are 190.34, 201.01, 213.21 and 226.8 for

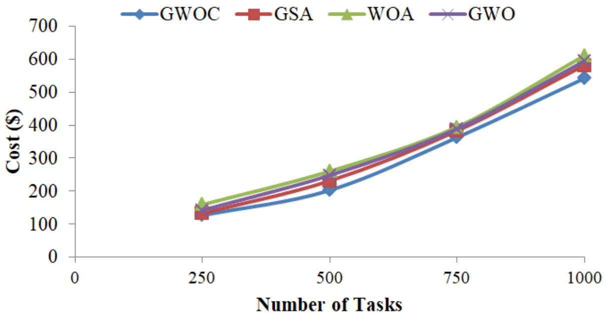

Figure. 4 cost of 100 VMs

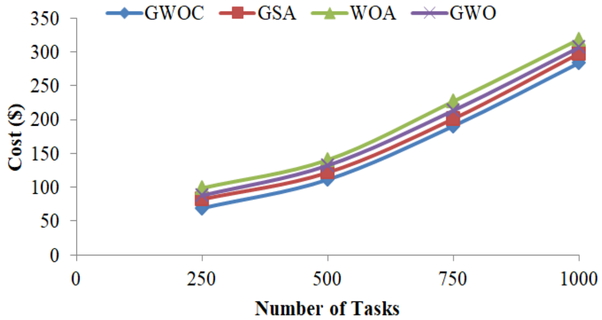

Figure. 5 cost of 200 VMs

GWOC, GSA, WOA and GWO algorithms respectively. Finally for 1000 tasks, the GWOC, GSA, WOA and GWO algorithms had produced values of 283.5, 297.44, 306.42 and 318.4 respectively. In all these cases, it could be seen that the proposed GWOC approach is clearly ahead of the other algorithms with respect to the cost parameter.

Hence it can be observed that the proposed GWOC technique had produced an improvement of 7.1%, 11.3% and 15.4% for cost when compared with GSA, WOA and GWO algorithms respectively, when used with 100 VMs. Also it had produced an improvement of 7.34%, 13.03% and 19.8% for makespan when compared with GSA, WOA and GWO algorithms respectively, when used with 200 VMs.

6. Conclusion

In this paper we had proposed a hybrid algorithm named grey wolf optimizer cuckoo (GWOC) to optimize the task scheduling activity in cloud. The proposed GWOC had been designed by integrating the grey wolf optimization and cuckoo search algorithms. It is known fact that multi-objective optimization approaches fare well in providing optimized performance when compared to single objective functions. In the simulated experiments that were carried out, performance of the GWOC had been evaluated based on the makespan and cost parameters. The task numbers were varied from 100 to 1000 for both the parameters. Also, results were obtained by considering 100 VMs and 200 VMs for both the parameters. The results obtained suggest that the proposed GWOC clearly outperforms the gravitational search algorithm (GSA), whale optimization algorithm (WOA), grey wolf optimizer (GWO) algorithms with respect to makespan and cost parameters. In future, the GWOC could be extended to real time applications by including additional QoS parameters.

Conflicts of interest

The authors declare no conflict of interest

Author contributions

Conceptualization, Pradeep Krishnadoss; methodology, Manikandan Nanjappan; software, Pradeep Krishnadoss; validation, Pradeep Krishnadoss; formal analysis, Gobalakrishnan Natesan; investigation, Gobalakrishnan Natesan; resources, Javid Ali; data curation, Javid Ali; writing— Javid Ali; writing—review and editing, Javid Ali; visualization, Pradeep Krishnadoss; supervision, Balasundaram Ananthakrishnan.

References

[1] L. Zuo, L. Shu, S. Dong, C. Zhu and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing", *IEEE Access*, Vol. 3, pp. 2687-2699, 2009.

[2] M. Abdullahi and M. A. Ngadi, "Symbiotic organism search optimization based task scheduling in cloud computing environment", *Future Generation Computer Systems*, Vol. 56 pp. 640-650, 2016.

[3] X. Zuo, G. Zhang, and W. Tan, "Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud", *IEEE Transactions on Automation Science and Engineering*, Vol. 11, No. 2, pp. 564-573, 2013.

[4] L. J. Ali and G. S. Anandhamala, "Actor-oriented approach for fault tolerance in e-commerce events", *Cluster Computing*, Vol. 21, No.1, pp. 239–250, 2018.

[5] R. Medara, R. S. Singh, and M. Sompalli, "Energy and cost aware workflow scheduling in clouds with deadline constraint", *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 13, p. e6922, 2022.

[6] D. Saxena, I. Gupta, J. Kumar, and A. K. Singh, "A secure and multiobjective virtual machine placement framework for cloud data center", *IEEE Systems Journal*, Vol. 16, No. 2, pp. 3163-3174, 2022.

[7] N. P. Marri and N. R. Rajalakshm, "MOEAGAC: an energy aware model with genetic algorithm for efficient scheduling in cloud computing", *International Journal of Intelligent Computing and Cybernetics*, Vol. 15 No. 2, pp. 318-329, 2022.

[8] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh, and A. Zomaya, "ARPS: An autonomic resource provisioning and scheduling framework for cloud platforms", *IEEE Transactions on Sustainable Computing*, Vol. 7, No. 2, pp. 386-399, 2021.

[9] M. Kaur and R. Aron, "Focalb: Fog computing architecture of load balancing for scientific workflow applications", *Journal of Grid Computing*, Vol. 19, No. 4, pp. 1-22, 2021.

[10] K. Dubey and S. C. Sharma, "A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing", *Sustainable Computing: Informatics and Systems*, Vol. 32, p.100605, 2021.

[11] M. S. Ajmal, Z. Iqbal, F. Z. Khan, M. Ahmad, I. Ahmad, and B. B. Gupta, "Hybrid ant genetic algorithm for efficient task scheduling in cloud data centers", *Computers and Electrical Engineering*, Vol. 95, p. 107419, 2021.

[12] Chhabra, G. Singh, and K. S. Kahlon, "QoS-aware energy-efficient task scheduling on HPC cloud infrastructures using swarm-intelligence meta-heuristics", *Computers, Materials & Continua*, Vol. 64, No. 2, pp. 813-834, 2020.

[13] R. Masadeh, N. Alsharman, A. Sharieh, B. A. Mahafzah, and A. Abdulrahman, "Task scheduling on cloud computing based on sea lion optimization algorithm", *International Journal of Web Information Systems*, Vol. 17 No. 2, pp. 99-116, 2021.

[14] K. Sreenu and S. Malempati, "MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing", *IETE Journal of Research*, Vol. 65, No. 2, pp. 201-215, 2019.

[15] K. Pradeep, L. J. Ali, N. Gobalakrishnan, C. J. Raman, and N. Manikandan, "CWOA: hybrid approach for task scheduling in cloud environment", *The Computer Journal*, Vol. 65, No. 7, pp. 1860-1873, 2022.

[16] N. K. CH, S. R. Inkollu, R. Patil, and V. Janamala, "Aquila Optimizer Based Optimal Allocation of Soft Open Points for Multi-Objective Operation in Electric Vehicles Integrated Active Distribution Networks",

International Journal of Intelligent Engineering and Systems, Vol. 14, No. 4, pp. 269-278, 2022, doi: 10.22266/ijies2022.0831.25.

[17] P. D. Kusuma and A. L. Prasasti, "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179-190, 2022, doi: 10.22266/ijies2022.1231.18.

[18] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[19] S. Ummadisetty and M. Tatineni, "Automatic Atrial Fibrillation Detection Using Modified Moth Flame Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 435-445, 2022, doi: 10.22266/ijies2023.0228.38.

[20] C. R. Rao, R. Balamurugan, R. Alla, "Artificial Rabbits Optimization Based Optimal Allocation of Solar Photovoltaic Systems and Passive Power Filters in Radial Distribution Network for Power Quality Improvement", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 100-109, 2022, doi: 10.22266/ijies2023.0228.09.

[21] M. Manohara, V. C. V. Reddy, and M. V. Kumar, "Northern Goshawk Optimization for Optimal Allocation of Multiple Types of Active and Reactive Power Distribution Generation in Radial Distribution Systems for Techno-Environmental Benefits", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 91-99, 2023, doi: 10.22266/ijies2023.0228.08.

[22] N. D. S. S. Kiran Relangi, A. Chaparala, R. Sajja, "Effective Groundwater Quality Classification Using Enhanced Whale Optimization Algorithm with Ensemble Classifier", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 214-223, 2023, doi: 10.22266/ijies2023.0228.19.

[23] P. D. Kusuma and A. Dinimaharawati, "Fixed Step Average and Subtraction Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 339-351, 2022, doi: 10.22266/ijies2022.0831.31.

[24] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle optimization algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273-281, 2022, doi: 10.22266/ijies2022.0228.25.

[25] R. K. Khadanga, A. Kumar, and S. Panda, "A modified Grey Wolf Optimization with Cuckoo Search Algorithm for load frequency controller design of hybrid power system", *Applied Soft Computing*, Vol. 124, p. 109011, 2022.

[26] P. Krishnadoss, N. Pradeep, J. Ali, M. Nanjappan, P. Krishnamoorthy, and V. K. Poornachary, "CCSA: Hybrid cuckoo crow search algorithm for task scheduling in cloud computing", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 241-250, 2021, doi: 10.22266/ijies2021.0831.22.

[27] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, "A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing", *Future Generation Computer Systems*, Vol. 83, pp. 14-26, 2018.

[28] B. H. A. Alguni, N. A. Alawad, "Distributed Grey Wolf Optimizer for scheduling of workflow applications", *Applied Soft Computing*, Vol. 102, pp. 14-26, 2021.