International Journal of Technology 10(1): 126-136
ISSN 2086-9614

OPTIMAL TASK SCHEDULING IN THE CLOUD ENVIRONMENT USING A MEAN GREY WOLF OPTIMIZATION ALGORITHM

Gobalakrishnan Natesan[1*], Arun Chokkalingam[2]

[1]*Research Scholar, Sathyabama Institute of Science and Technology, Department of Information Technology, St.Joseph's College of Engineering, Chennai, Tamil nadu 600119, India*
[2]*R.M.K College of Engineering & Technology, Chennai, Tamil nadu 601206, India*

(Received: May 2018 / Revised: June 2018 / Accepted: July 2018)

ABSTRACT

Cloud computing is one of the emerging areas in computing platforms, supporting heterogeneous, parallel and distributed environments. An important challenging issue in cloud computing is task scheduling, which directly influences system performance and its efficiency. The primary objective of task scheduling involves scheduling tasks related to resources and minimizing the time span of the schedule. In this study, we propose a Modified Mean Grey Wolf Optimization (MGWO) algorithm to enhance system performance, and consequently reduce scheduling issues. The main objective of this method is focused upon minimizing the makespan (execution time) and energy consumption. These two objective functions are elaborated in the algorithm in order to suitably regulate the quality of results based on response, in order to achieve a near optimal solution. The implementation results of the proposed algorithm are evaluated using the CloudSim toolkit for standard workloads (normal and uniform). The advantage of the proposed method is evident from the simulation results, which show a comprehensive reduction in makespan and energy consumption. The outcomes of these results show that the proposed Mean GWO algorithm achieves a 8.85% makespan improvement compared to the PSO algorithm, and 3.09% compared to the standard GWO algorithm for the normal dataset. In addition, the proposed algorithm achieves 9.05% and 9.2% improvement in energy conservation compared to the PSO and standard GWO algorithms for the uniform dataset, respectively.

Keywords: Cloud computing; Energy; Grey Wolf Optimization; Makespan; Optimization

1. INTRODUCTION

Cloud computing is one of the most aggressively competent technologies in the field of computing. It enables companies to maximize their productivity potential, pushing them towards the excellence of providing a unified service for their respective customers. Each of the fortune 500 listed companies is transferring their infrastructure in to the cloud domain. Cloud computing requires internet connection through which any user can store their data remotely and access the same form somewhere else. (Buyya et al., 2009). One of the core reasons why cloud computing is as popular as it is, is that it is scalable, reliable and allows end users to focus on the product, rather than worrying about how to deploy the service, which now is often managed by third party companies. The following are the services that are currently being offered under the cloud domain: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Virtualized applications are developed for offering such services through the

[*]Corresponding author's email: gobalakrishnanse@gmail.com, Tel. +91-9789708426
Permalink/DOI: https://doi.org/10.14716/ijtech.v10i1.1972

Internet. (Zhang et al., 2010; Pradeep & Jacob, 2018; Jennings & Stadler, 2015; Mustafa et al., 2015).

The unique selling point of cloud services is that they are flexible, dynamic and most importantly reduce the possibilities of degradation in performance. This is the main focus of various researchers around the world, with active fields including security, task scheduling, privacy concerns, cloud performance and deployment (Ma et al., 2014; Kumar & Sharma, 2017). As users are using shared resources in the cloud environment, it is important that task scheduling is performed in an efficient manner. User requirements are fulfilled by mapping their need with required resources with appropriate algorithms. Efficient solutions are provided to users by making use of more complex efficient scheduling algorithms (Dong et al., 2015). Problems could be a few or more, depending upon the complexity nature of applications. More complex applications require efficient algorithms to manage the data centers.

As the mapping of user tasks to the cloud resource is a complex problem, it is important to make use of optimization techniques to find near optimal solutions. Solutions for problems falling under NP-hard category can be obtained through enumeration techniques, heuristic or approximation method. The enumeration approach is usually not preferred because of its time consuming nature as it involves the building of all possible task schedulers that are needed and subsequently comparing each of them to arrive at the best solution (Gobalakrishnan & Arun, 2016). This leads us towards the next available option of heuristic and meta-heuristic techniques. Heuristic search techniques generate feasible solutions at a high operating cost, so these can be safely ignored (Shi et al., 2017; Natesan & Chokkalingam, 2018; Pradeep & Jacob, 2018). Therefore the only realistic possible solution could be obtained by adopting the meta-heuristic algorithm approach, such as the Genetic Algorithm (GA) (Gutierrez-Garcia & Sim, 2012), Particle Swarm Optimization (PSO) (Poli et al., 2007), or Grey Wolf Optimization (GWO) (Mirjalili et al., 2014), which employ a pool of candidate solutions to traverse the available solution space. It is also understood that GWO falls under the category of swarm intelligence (Pacini et al., 2014; Ghomi et al., 2017; Singh & Chana, 2016). In addition, factors such as adaptive exploratory behavior, little controlling parameter have motivated the use of GWO in this research work. In the paper, we propose a cloud task scheduling algorithm based on a metaheuristic Mean Grey Wolf Algorithm to minimize energy consumption and the execution time of the task in the cloud. Execution of the proposed algorithm is carried out in the CloudSim environment, for common workloads in simulated data centers.

2. RELATED WORK

Task scheduling prevails as a hefty defiance in the cloud computing sector, considering the heterogeneous resources. In recent years, many researchers have contemplated metaheuristic algorithms to solve the NP-hard problem (Duan et al., 2014; Sadhasivam & Thangaraj, 2017; Natesan & Chokkalingam, 2017; Pradeep & Jacob, 2017). Moon et al. (2017) have formulated the slave ants-based Ant Colony Optimization algorithm for scheduling user tasks in a heterogeneous environment. Their research work mainly focuses on minimizing the task execution time. They also adapted diversification and reinforcement strategies, whose pheromones are wrongly accumulated by helping ants to avoid long paths. Pacini et al. (2015) have proposed a scientific application scheduling scheme, which takes into account the balancing of throughput and response time. They used an ACO algorithm to address the scientific application scheduling problems in the IaaS cloud environment.

Kaur and Chana (2016) presented energy aware scheduling on deadline constrained tasks in a multi objective cloud environment. Energy efficiency is ensured by considering the virtualization and consolidation technique in a multi-core processor. Their work utilized the green cloud scheduling model (GCSM), which exploits tasks and resources with heterogeneity. The scheduler schedules and allocates deadline-constrained tasks which only impose energy-intended nodes. They achieved 71% of energy savings, resulting in high performance by meeting deadlines. Ramezani et al. (2015) achieved multi-objective task scheduling optimization for NP-hard problems by reducing cost and response time and promoting Quality of Service. They considered four objectives, namely minimizing the task execution cost, task queue length, task transfer time and power consumption, in order to enhance the multi-objective optimization model. They proposed two algorithms to resolve the optimal task arrangement. Zhou and Dong (2018) presented hybrid glowworm swarm optimization (HGSO) for scheduling tasks in order to improve performance. They used a quantum behavior strategy, which depends on the principles of neighborhood, production, random walk and offspring. All these issues are taken into account to produce efficient scheduling at a favorable cost. An independent task was used to reduce dependencies in order to achieve an optimal solution. Khalili and Babamir (2017) proposed a pareto-based Grey Wolf Optimizer to schedule workflow tasks in a cloud environment, which helps to minimize task completion time and VM usage cost and to maximize resource throughput.

3. METHODS

3.1. Proposed Framework

The proposed system framework is shown in Figure 1. The task manager or cloud broker component aggregates and manages the tasks submitted by users or clients. All such collected tasks are acquired by the cloud broker and presented to the scheduler. The user submitted tasks are then assigned to appropriate virtual machines by the scheduler. The scheduler works at an equivalent level with the Resource Information Server (RIS) to optimize the assignment process. The scheduler gets the knowledge of the available resources and capabilities of each VM through the Resource Information Server. Summarized information obtained from the data center is presented by the RIS. The data center manages an aggregated collection of cloud resources that includes both virtualized and physical resources, together with CPU, memory and storage resources. Upon knowing the availability of resources, the scheduler performs an analysis and chooses the appropriate VM to whom the task could be assigned. This leads to increased efficiency in the task scheduling process. The VMs are predefined, and tasks are allocated by the scheduler before the execution of the task scheduling process.

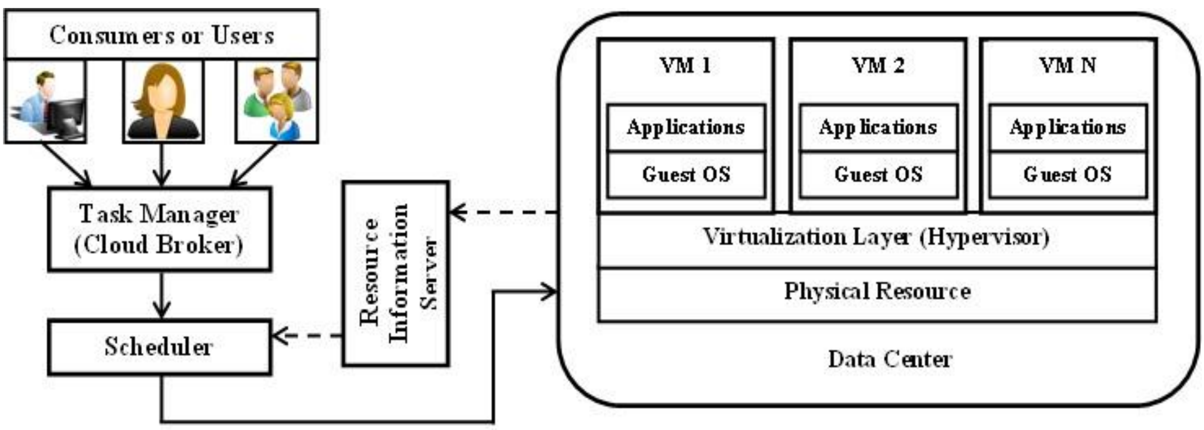


Figure 1 Task scheduling framework in the cloud

3.2. Problem Description

The cloud information server acquires information about the available resources in the data center that are necessary to complete the user requested tasks. These tasks are directed to the task manager, which is also referred to as the Cloud Broker (CB). For example, let us assume that the tasks (T_1, T_2, T_3,…….,T_n) are presented to the cloud broker over an interval of time, that the processing elements (the VMs) have individual processing rates, and that memory may vary. It is vital to note that executing the tasks on different VMs would result in different execution times and costs. If VMs (VM_1, VM_2, ….VM_m) are available for processing, the tasks sent to the CB are scheduled by the scheduler on the appropriate ones. The aim is to assign the tasks on the VMs by making optimal use of their minimal makespan. The method proposed achieves this aim by making use of the Expected Time to Complete (ETC) parameter of each VM and then schedules tasks accordingly. The ETC values are computed using the Million Instructions per Second (MIPS) of the VM to the task length ratio. These calculated ETC values are represented in matrix form, in which the columns denote the number of available VMs and the rows denote the number of tasks to be scheduled. In general, each row and column of the matrix represents the execution times of a given task for each VM and of each task on a given VM respectively.

3.3. Performance Metrics

3.3.1. Makespan

The objective of the proposed approach is to reduce the makespan by choosing the appropriate VMs to whom tasks could be assigned. Let ($\{1,2,3\ldots C_{xy}, x\in m\}, y\in\{1,2,3\ldots,n\}$) be the execution time for executing yth numbered task on the chosen xth VM, where 'x' is the number of VMs and 'y' denotes the number of tasks. Then the fitness value for makespan can be computed from the below equation:

$$Makespan = max\{C_{xy}, \forall all\ tasks\ y\ mapped\ to\ VM_i\} \quad (1)$$

$$i \in \{1, 2, 3 \ldots \ldots, m\}$$

$$F_{MP}^{min} = max\{C_{xy}\} \quad (2)$$

3.3.2. Energy consumption

The final objective of the proposed approach is account of energy consumed. Energy consumed depends upon the number of data centers present or available in the cloud. An increase in the number of data centers leads to an increase of energy consumed. The energy consumed can be obtained by multiplying the elements of the distribution matrix with the energy consumed by the virtual machine, which is given by:

$$EC = \frac{1}{VM \times T} \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} \times d_{ij} \quad (3)$$

where r_{ij} represents an element in the distribution matrix, and d_{ij} is the energy utilized by each resource during the execution of a task. VM denotes virtual machine and T represents the number of tasks. The energy consumption objective function is represented by:

$$F_{EC}^{min} = EC \quad (4)$$

3.3.3. Fitness function

The main objective of the proposed approach is to schedule tasks in the cloud such that it results in the optimization of execution time and energy consumed. The objective is achieved by deducing a fitness function through which the quality of the obtained solutions can be evaluated. But as for each optimization problems, only a nearest optimal solution could be obtained from the deduced fitness function. The fitness function for evaluating time and energy parameters is given in Equation 5:

$$FF\ (I) = \gamma \times F_{MP}^{min} + \varphi \times F_{EC}^{min} \quad (5)$$

where the fitness function is denoted as *FF (I)*. Here, γ and φ are constant value, and the value of γ, φ are between 1> γ, φ ≥0. F_{MP}^{min} is the first objective, to minimize the task execution time, and F_{EC}^{min} is the second objective, to minimize energy consumption.

3.4. Modified Mean Grey Wolf Optimization Algorithm (MGWO)

A Mean Grey Wolf Optimization variant algorithm has been proposed in this work to increase the accuracy and performance of the GWO algorithm. GWO employs a pool of candidate solutions to traverse the available solution space. The social leadership and hunting techniques among the grey wolves is simulated by using the GWO algorithm. According to this algorithm, the grey wolf colony is partitioned into four divisions, namely α, β, δ and ω. In each processing, the first three best candidate solutions are α, β and δ. The remaining grey wolves are tagged as ω. Both the encircling as well as hunting equations have been modified in the proposed approach. All other remaining equations, as well as procedures, are similar to that present in the standard GWO algorithm (Mirjalili et al., 2014). The main purpose of the proposed technique is to increase the efficiency of the motion and suitable path of each wolf present in the search area. The Mean GWO algorithm is focussed in the following parts:

Encircling prey:

During the hunt, the prey that is encircled by the grey wolves can be improvised using the following equations:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \theta \cdot \left(\vec{X}(t)\right)\right| \tag{6}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D} \tag{7}$$

where θ is the mean, the prey position vector is denoted as $\overrightarrow{X_p}$, 't' is the current iteration, and the grey wolf position vector is denoted as $\vec{X}(t)$.

Vectors $\vec{A}$ and $\vec{C}$ are denoted as:

$$\vec{A} = 2\vec{a} \cdot \overrightarrow{r_1} - \vec{a} \tag{8}$$

$$\vec{C} = 2 \cdot \overrightarrow{r_2} \tag{9}$$

where the element $\vec{a}$ decreases from 2 to 0. $\overrightarrow{r_1}$, $\overrightarrow{r_2}$ take certain random values lying between [0,1].

Hunting:

The α, β and δ groups irregularly participate and guide the hunting of the prey. Initially, the three best and optimal candidate solutions are depicted by α, β, δ, and the remaining solutions are denoted by ω. Each wolf's position has been improvised in the search space region by calculating the mean of the positions.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \theta \cdot \vec{X}(t)\right| \tag{10}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \theta \cdot \vec{X}(t)\right| \tag{11}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \theta \cdot \vec{X}(t)\right| \tag{12}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \left(\overrightarrow{D_\alpha}\right) \tag{13}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \left(\overrightarrow{D_\beta}\right) \tag{14}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \left(\overrightarrow{D_\delta}\right) \tag{15}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{16}$$

When the prey becomes idle, the grey wolves attack it and end their hunting. To calculate this through a mathematical model, the value of $\vec{a}$ decreases when the wolf approaches the prey. $\vec{A}$ is considered to be a value in the interval [-2a, 2a], where the coefficient ($\vec{a}$) decreases from 2 to 0, so due to this relation between them $\vec{A}$ also fluctuates. The proximity of the search agent lies between its current position and the position of the prey, when the values of $\vec{A}$ are considered to be between [-1, 1].

3.5. The Pseudo Code of the Mean Grey Wolf Optimization Algorithm (MGWO)

Input:

Parameters of Mean-GWO Algorithm.

Task (T_y) and Resource (R_x) $\forall\ x \in \{1,2,\ldots m\}$ *and* $y \in \{1, 2, \ldots..,n\}$.

Output:

The tasks are scheduled to an optimal resource $\{T_y, R_x\}$.

Parameter initialization:

1. 'n' is an original population size $\vec{X_i} = (i = 1, 2, \ldots \ldots . n)$, factor 'a', C & A coefficient vector and Max_{itr} maximum iteration.
2. Set counter preliminary value t=0

Population initialization:

3. Set i = 1
4. **While** ($i \leq n$) **do**
5. Randomly generate initial population $\overrightarrow{X_i(t)}$
6. Compute the fitness of every single grey wolf in the search area using Equation (5)
7. **End While**
8. The grey wolf with most fitness is denoted as Alpha $\vec{X_\alpha}$.
9. The grey wolf with the second most fitness is denoted as Beta $\vec{X_\beta}$.
10. The grey wolf with the third most fitness is denoted as Delta $\vec{X_\delta}$.

Solution Updating phase

11. **While (t<Maxitr)** //Maximum number of iterations
12. **For** each search agent
13. Using equation (16) each grey wolf position is updated.
14. **End for**
15. The value of 'a' is reduced from 2 to 0.
16. Using Equations (8) and (9), the coefficients of "A" and "C" are updated respectively.
17. Compute the fitness value of all grey wolves using Equation (5).
18. Update the positions of Alpha ($\vec{X_\alpha}$), Beta ($\vec{X_\beta}$) and Delta ($\vec{X_\delta}$) using equations (13), (14) and (15).
19. set t=t+1 //increasing the iteration counter
20. **End while**

Best Solution

21. return Alpha ($\vec{X_\alpha}$) as the nearest optimal solution from the search space.

4. RESULTS AND DISCUSSION

4.1. Experimental Setting

This section concerns the computational experiments that were performed to evaluate the performance of the proposed technique. The proposed algorithm was simulated using the CloudSim tool; the base platform for this toolkit relies on JAVA. All the experiments were validated on a PC with Intel(R) Core(TM) i5-457, 4 CPU @ 2.9 GHz, RAM of 8GB and 64-bit Windows OS as its configuration. The simulation results that are obtained using the proposed approach with respect to consumption of energy and time are discussed in this section and the obtained results are compared with PSO and standard GWO approaches. The parameters are represented in the form of table, with the parameter setup of VMs and tasks displayed in Table 1. The performance of the proposed technique is evaluated by considering two different datasets. They are normal and uniform (Abdullahi et al., 2016) and each distribution has 100-500 tasks. The normal dataset has medium sized tasks and fewer small and large sized tasks; whereas the uniform one has equal number of small, medium and large sized tasks.

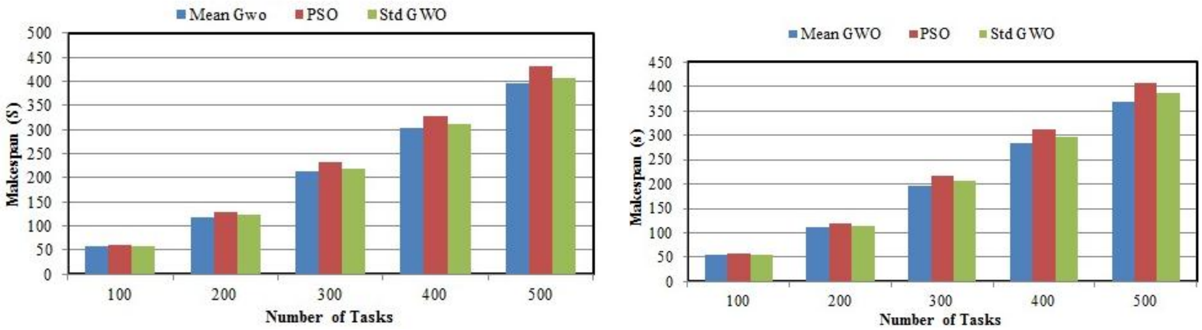
Table 1 Experimental settings

No	Parameter	Value
1	Number of Data Center	1
2	Number of Host	10
3	Host Memory Capacity	10 GB
4	Host Bandwidth	2800 Mbps
5	Number of VMs	50
6	VM Policy	Time_Shared
7	VMM/Hypervisor	Xen
8	Number of vCPU	[1 – 5]
9	VM RAM Size	[512 – 2048] MB
10	vCPU Capacity	[500 – 2500] MIPS
11	VM Operating System	Linux
12	Bandwidth	[250 – 1500]
13	Number of Tasks	[100 – 500]
14	Task MIPS	[200 – 15000]

4.2. Evaluation of Makespan

The performance results of the proposed mean-GWO approach are compared with PSO and standard GWO with respect to makespan for the task scheduling process. The datasets used are the normal and uniform distribution, and the number of tasks varied from 100–500. The simulation is performed by executing the tasks 30 times and an average was deduced from it. 50 VMs were considered for all the datasets. Figures 2a and 2b depict the average makespan for the executed tasks obtained by applying the proposed mean-GWO, PSO and standard GWO. For the normal distribution, it could be observed that when there are 100 cloud tasks, the makespan values for mean-GWO, PSO and standard GWO are 56.27, 61.25 and 58.01, respectively. Similarly, when there are 500 tasks, the makespan values for mean-GWO, PSO and standard GWO are 395.66, 430.68 and 407.90, respectively. The Performance Improvement Rate (PIR%) for the normal distribution makespan value of the proposed mean-GWO, PSO and standard GWO is shown in Table 2. The PIR of the normal distribution shows that the proposed mean-GWO achieves 8.85% and 3.09% makespan improvement over PSO and standard GWO, respectively.
Similarly, in the uniform distribution, when there are 200 cloud tasks, the makespan values are 111.73, 120.02 and 114.02. When there are 400 cloud tasks, the makespan values are 284.21, 312.35 and 296.73. In the uniform distribution, the makespan improvements over PSO and standard GWO are 9.67 and 4.188, respectively. These makespan improvement values are shown

in Table 3. Hence it could be observed that the makespan values obtained through mean-GWO are far better than those obtained through PSO and standard GWO.



(a) (b)

Figure 2 (a) Makespan of Normal for 50 VMs; (b) Makespan of Uniform for 50 VMs

Table 2 Normal distribution PIR (%) for 50 VMs

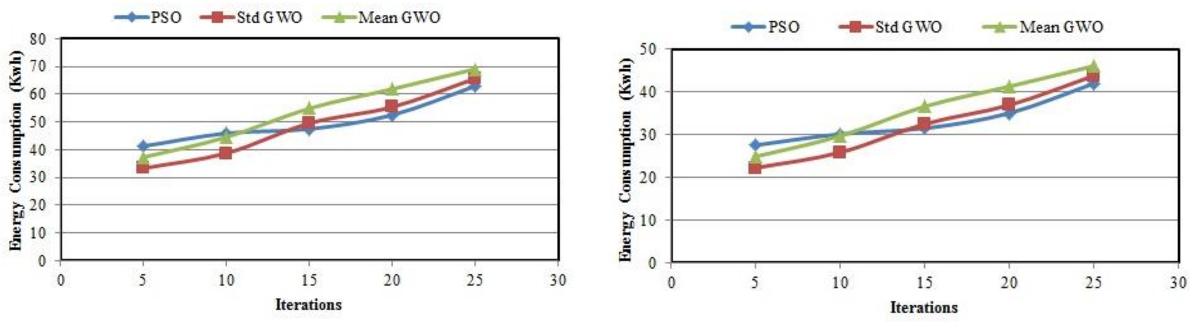
	PSO	Standard GWO	Mean GWO
Total Makespan	1180.85	1118.392	1084.84013
PIR % over PSO		5.584642	8.85014663
PIR % over Std-GWO			3.0927

Table 3 Uniform distribution PIR (%) for 50 VMs

	PSO	Standard GWO	Mean GWO
Total Makespan	1113.25	1057.589	1015.07187
PIR % over PSO		5.263158	9.67219715
PIR % over Std-GWO			4.188

4.3. Energy Consumption

Figures 3a and 3b depict the energy consumption of the normal and uniform distribution datasets using mean-GWO, PSO and standard GWO. From the experimental results, the proposed mean-GWO technique outperforms the existing algorithms of PSO and standard GWO in terms of energy consumption. Simulation was carried out for 25 iterations to evaluate the energy consumed while executing the tasks. It could be observed that better results are produced by PSO technique when the iterations are less in the range of (5, 10) than other two techniques. But when the iteration count is increased, the proposed mean–GWO method produces better results than the PSO and standard GWO techniques. Energy consumed while executing 400 tasks with various iterations are shown in Figure 3a. Figure 3b depicts the uniform data energy consumed for 400 tasks with various iterations. The experimental analysis shows that the proposed technique consumes comparatively minimum energy for the normal and uniform datasets during task execution.



(a) (b)

Figure 3 (a) Energy consumption of Normal for 400 tasks; (b) Energy consumption of uniform for 200 tasks

5. CONCLUSION

The proposed approach fulfills the main objective of task scheduling by optimizing the execution time and energy consumed while executing the tasks in the cloud environment. A Mean-GWO technique has been proposed in this work to achieve this objective. The proposed technique has been evaluated with two datasets (normal and uniform distribution). The results obtained from the simulation carried out using the proposed mean-GWO technique clearly shows an improvement in the performance of task scheduling process when compared with the existing PSO and standard GWO techniques. In future, task scheduling can be attained in federated data centers based on the bio-inspired technique (Pacini et al., 2016; Aruna & Aramudhan, 2016). In addition, apart from QoS parameters such as execution time and energy, other parameters such as reliability, load imbalance and security can also be integrated with the Mean-GWO technique and implemented in the real cloud environment.

6. REFERENCES

Abdullahi, M., Ngadi, M.A., Abdulhamid, S.M., 2016. Symbiotic Organism Search Optimization Based Task Scheduling in Cloud Computing Environment. *Future Generation Computer Systems*, Volume 56, pp. 640–650

Aruna, L., Aramudhan, M., 2016. Framework for Ranking Service Providers of Federated Cloud Architecture using Fuzzy Sets. *International Journal of Technology*, Volume 7(4), pp. 643–653

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2009. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems*, Volume 25(6), pp. 599–616

Dong, Z., Liu, N., Rojas-Cessa, R., 2015. Greedy Scheduling of Tasks with Time Constraints for Energy-efficient Cloud-computing Data Centers. *Journal of Cloud Computing*, Volume 4(1), pp. 1–14

Duan, R., Prodan, R., Li, X., 2014. Multi-objective Game Theoretic Scheduling of Bag-of-tasks Workflows on Hybrid Clouds. *IEEE Transactions on Cloud Computing*, Volume 2(1), pp. 29–42

Gobalakrishnan, N., Arun, C., 2016. SIS: A scheme for dynamic independent task scheduling in a cloud environment. International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT-2016), pp. 272-279,

Ghomi, E.J., Rahmani, A.M., Qader, N.N., 2017. Load-balancing Algorithms in Cloud Computing: A Survey. *Journal of Network and Computer Applications*, Volume 88, pp. 50–71

Gutierrez-Garcia, J.O., Sim, K.M., 2012. GA-based Cloud Resource Estimation for Agent-based Execution of Bag-of-tasks Applications. *Information Systems Frontiers*, Volume 14(4), pp. 925–951

Jennings, B., Stadler, R., 2015. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*, Volume 23(3), pp. 567–619

Kaur, T., Chana, I., 2016. Energy Aware Scheduling of Deadline-constrained Tasks in Cloud Computing. *Cluster Computing*, Volume 19(2), pp. 679–698

Khalili, A., Babamir, S.M., 2017. Optimal Scheduling Workflows in Cloud Computing Environment using Pareto-based Grey Wolf Optimizer. *Concurrency and Computation: Practice and Experience*, Volume 29(11), pp. 1–15

Kumar, M., Sharma, S.C., 2017. Dynamic Load Balancing Algorithm to Minimize the Makespan Time and Utilize the Resources Effectively in Cloud Environment. *International Journal of Computers and Applications*, Volume 39, pp. 1–10

Ma, T., Chu, Y., Zhao, L., Ankhbayar, O., 2014. Resource allocation and scheduling in cloud computing: policy and algorithm. *IETE Technical review*, Volume 31(1), pp. 4–16

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey Wolf Optimizer. *Advances in Engineering Software*, Volume 69, pp. 46–61

Moon, Y.J., Yu, H., Gil, J.M., Lim, J.B., 2017. A Slave Ants Based Ant Colony Optimization Algorithm for Task Scheduling in Cloud Computing Environments. *Human-centric Computing and Information Sciences*, Volume 7(28), pp. 1–10

Mustafa, S., Nazir, B., Hayat, A., Khan, A.U.R., Madani, S.A., 2015. Resource Management in Cloud Computing: Taxonomy, Prospects, and Challenges. *Computers & Electrical Engineering*, Volume 47, pp. 186–203

Natesan, G., Chokkalingam, A., 2018. A New Multi-objective Optimal Programming Model for Task Scheduling using Genetic Gray Wolf Optimization in Cloud Computing. *The Computer Journal*, Volume 61(10), pp. 1523–1536

Natesan, G., Chokkalingam, A., 2017. Opposition Learning-based Grey Wolf Optimizer Algorithm for Parallel Machine Scheduling in Cloud Environment. *International Journal of Intelligent Engineering and Systems*, Volume 10(1), pp. 186–195

Pacini, E., Mateos, C., Garino, C.G., 2014. Distributed Job Scheduling based on Swarm Intelligence: A Survey. *Computers & Electrical Engineering*, Volume 40(1), pp. 252–269

Pacini, E., Mateos, C., Garino, C. G., 2015. Balancing Throughput and Response Time in Online Scientific Clouds via Ant Colony Optimization (SP2013/2013/00006). *Advances in Engineering Software*, Volume 84, pp. 31–47

Pacini, E., Mateos, C., Garino, C.G., Careglio, C., Mirasso, A., 2016. A Bio-inspired Scheduler for Minimizing Makespan and Flowtime of Computational Mechanics Applications on Federated Clouds. *Journal of Intelligent & Fuzzy Systems*, Volume 31(3), pp. 1731–1743

Poli, R., Kennedy, J., Blackwell, T., 2007. Particle Swarm Optimization. *Swarm Intelligence*, Volume 1(1), pp. 33–57

Pradeep, K., Jacob, T.P., 2017. CGSA Scheduler: A Multi-objective-based Hybrid Approach for Task Scheduling in Cloud Environment. *Information Security Journal: A Global Perspective*, Volume 27(2), pp. 77–91

Pradeep, K., Jacob, T.P., 2018. OCSA: Task Scheduling Algorithm in Cloud Computing Environment. *International Journal of Intelligent Engineering and Systems*, Volume 11(3), pp. 271–279

Pradeep, K., Jacob, T.P., 2018. A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications*, Volume 101(4), pp. 2287-2311.

Ramezani, F., Lu, J., Taheri, J., Hussain, F.K., 2015. Evolutionary Algorithm-based Multi-Objective Task Scheduling Optimization Model in Cloud Environments. *World Wide Web,* Volume 18(6), pp. 1737–1757

Sadhasivam, N., Thangaraj, P., 2017. Design of an Improved PSO Algorithm for Workflow Scheduling in Cloud Computing Environment. *Intelligent Automation & Soft Computing,* Volume 23(3), pp. 493–500

Singh, S., Chana, I., 2016. A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges. *Journal of Grid Computing*, Volume 14(2), pp. 217–264

Shi, L., Zhang, Z., Robertazzi, T., 2017. Energy-aware Scheduling of Embarrassingly Parallel Jobs and Resource Allocation in Cloud. *IEEE Transactions on Parallel and Distributed Systems*, Volume 28(6), pp. 1607–1620

Zhang, Q., Cheng, L., Boutaba, R., 2010. Cloud Computing: State-of-the-art and Research Challenges. *Journal of Internet Services and Applications*, Volume 1(1), pp. 7–18

Zhou, J., Dong, S., 2018. Hybrid Glowworm Swarm Optimization for Task Scheduling in the Cloud Environment. *Engineering Optimization*, Volume 50(6), pp. 949–964