

# Workflow Scheduling using Improved Moth Swarm Optimization Algorithm in Cloud Computing

**Michael Mahesh K**

Asst. Professor,  
St. Joseph College of Engineering, Chennai, Tamil Nadu, India  
michaelmahesh.k97@gmail.com

**Abstract:** Cloud computing is the advancing technology that aspires at providing services to customers with the available resources in the cloud environment. While the cloud server receives the multiple user requests, it is essential for the appropriate resources scheduling to accomplish appropriate customer fulfillment. Hence, this work exploits a developed Improved Moth Swarm Optimization (IMSO) method to carry out multi-objective task scheduling in the cloud environment. The developed method that schedules tasks based on a fitness parameter. The fitness parameter based upon the three most important constraints such as the Quality of Service (QoS), energy, and resources utilization. The developed approach schedules the task based on the three constraints like cost and task execution time in virtual machines is low. A competent task scheduling based upon the minimum fitness parameter. The comparative analysis exploiting the investigational setups shows that the developed model carries out fine while evaluated with the conventional techniques.

**Keywords:** Cloud Computing; Multi-Objective; Workflow; Task Scheduling; Qos

## Nomenclature

Abbreviations	Descriptions
ANN	Artificial Neural Network
VMs	Virtual Machines
IoT	Internet of Things
WOA	Whale Optimization Algorithm
IaaS	Infrastructure as a Service
DNCPSO	Directional and Non-Local-Convergent PSO
ACO	Ant Colony Optimization
PM	Physical Machines
PSO	Particle Swarm Optimization
DMOP	Dynamic Multi-Objective Optimization Problem
IGS	Improved Greedy Search

## 1. Introduction

To present computing, network, and storage services, cloud computing has become an important force that has been extensively used in several fields in the last decades [10]. Nevertheless, using the rising demand for IoT, the framework of the cloud computing model is facing huge summon. At present, whereas smart IoT devices are ever-present, we live in such a world. These huge smart devices positioned at the edge of the network, which needs data processing like location-awareness, minimum-latency, and mobility necessities. To fulfill these demanding necessities edge computing is derived [1].

The drawbacks of cloud computing like communication latency can efficiently remunerate by this current computing paradigm, which is represented as “network edge cloud”[9]. Conversely, for end-user applications, edge computing can present the local computing, storage, and network that aid to ease the real-time interaction, location-awareness, mobility-support, and scalability.

Actually, the position of edge devices is near to end-user applications, hence it can operate the delay-sensitive tasks rapidly. Accordingly, edge computing is highly effectual by considering a few components like operating expenses, power utilization, service latency, network traffic, content distribution, etc. From this perspective, edge computing can superior to convene the application requires of IoT evaluated with cloud computing. Conversely, the scalability and flexibility of cloud computing can aid edge

computing so as to deal with the rising demands for large-scale computation-intensive business applications while the processing ability of edge devices is inadequate.

Workflows are the general methods to model the large scale compute and data-intensive applications from diverse research domains. In cloud computing, the workflow scheduling issue aspires to map the tasks of a presented application onto obtainable resources. It is an NP-complete issue, in that the orchestration of task executions is an important problem to optimize the objectives stated in QoS [10]. Workflow has often exploited to design large applications collected of dependent tasks. In scientific computing, the problem of scheduling workflows in conventional environments such as grids and clusters was extensively reported over several years. In recent times, using the extensive implementation of cloud computing, the research in workflow scheduling has been changing to executing workflows on IaaS platforms that can present on-demand VMs on a pay-as-you-go basis.

The objective of workflow scheduling is to discover appropriate mappings from the tasks to a set of parallel services (machines) that present computing resources. Fundamentally, there are two methods of exploiting services [11]. In the non-multi programmed employ, a scheduler can allocate only a single task to run on service at a time, while in the multiprogramming exploit, multiple tasks, which perchance positioned on the same service to run at the same time.

The main objective of the work is to propose an Improved Moth Flame approach that is the enhanced version of the conventional MOA. Here, the Fitness is represented as the important factor which is based upon three constraints, such as the energy, resource utilization, and the QoS. Moreover, the fitness factor is preserved at least by facilitating the utmost resource utilization, the least amount of energy, and the utmost QoS. The fitness model aspires at choosing the solution based on the least execution time and cost.

## 2. Literature Review

In 2019, S. Jaya Nirmala et al [1], proposed a novel fault-tolerant workflow scheduling approach that experiences the duplication heuristics in an unsupervised model. Also, the employ of lightweight synchronized checkpointing permits competent re-submission of unsuccessful tasks and assures workflow achievement even in insecure environments.

In 2020, Mutaz Barika et al [2], investigated the scheduling issue of stream workflow to hold up runtime modifications of stream workflow deployment, with the intention that scheduling plans can be amended to hold stream workflow applications using incessantly varying features. In a Multi-cloud environment, it proposes three diverse plug-in approaches and techniques to set-up auto-scaling of this workflow.

In 2019, Zhaomeng Zhu and Xueyan Tang [3], developed a list-scheduling model for the novel multi-programmed cloud resource model. This model slowly employs tasks the optimal positioning establishes on both conventional and novel VMs on the platform about a priority list. To completely use the heterogeneity of IaaS platforms, the VMs can be scaled up throughout the scheduling procedure. Subsequently, a deadline constrained workflow scheduling method named DyDL based on this model to optimize the workflow execution cost was proposed.

In 2020, Goshgar Ismayilov and Haluk Rahmi Topcuoglu [4], designed a dynamic workflow scheduling issue as a DMOP whereas the source of dynamism was based on both resource failures and the number of objectives that might change over time. A prediction based dynamic multi-objective evolutionary approach, named NN-DNSGA-II approach, by integrating ANN using the NSGA-II algorithm was proposed.

In 2020, Jin Sun, Lu Yin et al [5], developed relations of task allocation schemes to deal with diverse kinds of tasks in the workflow. Further, these schemes were combined with a greedy approach to build an IGS approach that was able to produce possible solutions fulfilling all restraints.

In 2019, Ying Xie et al [6], developed a new DNCPSO which uses non-linear inertia weight using selection and mutation operations through directional search procedure that can decrease makespan and cost radically also attain a promising result.

## 3. System Model

The system model of the cloud computing environment is presented in this section. The cloud environment comprises of numerous service providers to present services and infrastructure to the users as stated by their request. There are several PMs and every physical machine has several VMs in the cloud environment. In the cloud, these physical machines are obtainable to provide the services to the users. While multiple numerals of solicits enter the cloud soliciting for a similar resource, there subsists resources lack, and the issue is concerning the allocation of the resource to the users. Every VM has

several configurations such as memory, CPU size, and the execution cost of the tasks, etc. Hence, assigning the ideal VM which needs minimum time and cost to execute the task is considered as the most important problem. To attain the satisfaction of the customer, the workflow schedule comes into subsistence. The scheduling method orders the request to provide the service to the users. The schedule is on the basis of the energy and capacity obligatory to run the task. Let the cloud comprises of  $q$  number of PMs which is stated in eq. (1).

$$P = \{p_1, p_2, \dots, p_i, \dots, p_q\} \quad (1)$$

In eq. (1),  $p_1, p_2, \dots, p_i, \dots, p_q$  indicates the individual physical machines,  $|P|$  indicates the total count of PMs available in the cloud and  $p_i$  indicates the  $i^{\text{th}}$  PM in the cloud. The total VMs similar to  $i^{\text{th}}$  PM is stated in eq. (2).

$$|V^i| = \{V_1^i, V_2^i, \dots, V_j^i, \dots, V_g^i\} \quad (2)$$

In eq. (2),  $V^i$  indicate the total VMs similar to the  $i^{\text{th}}$  PM  $g$  indicate the total number of VMs and  $V_g^i$  indicate  $g$  number of VMs similar to the  $i^{\text{th}}$  PM.  $V_j^i$  indicate the  $j^{\text{th}}$  VM of  $i^{\text{th}}$  PM. The capacity of  $j^{\text{th}}$  VM present in  $i^{\text{th}}$  PM is indicated as  $C_j^i$ . The cost of the resource for  $j^{\text{th}}$  VM present in  $i^{\text{th}}$  PM is indicated as  $R_j^i$ .

In the cloud, assumes the total number of VMs is indicated as  $V$  and  $N$  number of VMs. For the resource, user requests to perform a definite task. Hence, the task length and the capability are the most important parameters whereas deciding the VM carries out the operation. Consider the total task is indicated as  $T$  and the total count of tasks is stated in eq. (3).

$$T = \{t_1, t_2, \dots, t_k, \dots, t_n\} \quad (3)$$

In eq. (3),  $n$  indicates the total count of tasks in the cloud,  $T$  indicates total tasks represent all individual tasks for implementation and  $t_k$  indicates  $k^{\text{th}}$  task. Every task takes a count of sub-tasks and the subtasks in  $k^{\text{th}}$  task are indicated in eq. (4)

$$t_k = \{t_1^k, t_2^k, \dots, t_l^k, \dots, t_m^k\} \quad (4)$$

In eq. (4),  $m$  indicate the total count of subtasks in  $k^{\text{th}}$  task and  $t_1^k, t_2^k, \dots, t_l^k, \dots, t_m^k$  indicate the subtasks of  $k^{\text{th}}$  task. The task length set up to determine the VM to execute the task. The task length is indicated as  $L$  and the length of the task-based upon the number of tasks waiting for implementation. One more parameter which determines VM for execution the task is energy. Consider the energy parameter is indicated as  $\varepsilon_{jk}$  that is the energy needed to execute the  $k^{\text{th}}$  task in the  $j^{\text{th}}$  VM.

#### 4. Proposed Method: Multi-Objective Task Scheduling Exploiting the RWO Approach:

The developed technique for scheduling the task that is based on the Improved Moth flame optimization algorithm is presented in this section. A count of user requests is satisfied based on their necessities in the cloud computing environment. However for the same resource while a number of users request, there subsists the shortage of resources for serving the users. Resources are abundant however there is a shortage of appropriate scheduling in the cloud. The proposed method execution time and the cost is minimum, and outcomes in an effectual task scheduling exploiting the proposed method. Fig. 1 demonstrates the developed IMOA algorithm. The task scheduling is based upon 3 constraints, such as the QoS, energy required to execute the task in the individual VM, and the utilization of the resource. Hence, the schedule method fulfills all the three constraints whilst assigning the VMs upon multiple user requests.

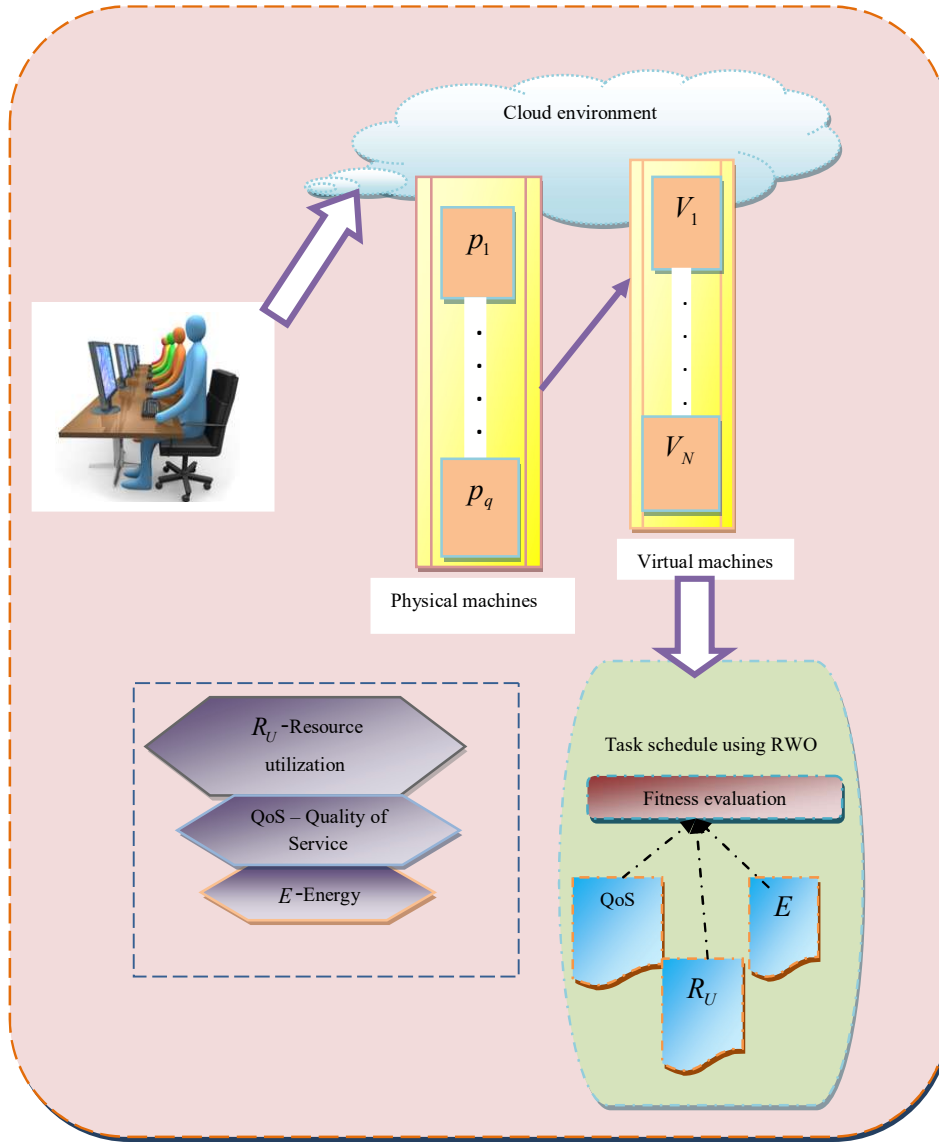


Fig. 1. System model of the proposed IMO A

## 5. Multi-Objective Scheduling

During a similar time, while the users request for a similar resource, the requirement for task scheduling happens and scheduling approach based upon the fitness parameter of the solution matrix. A competent solution is decided during the estimate of the fitness which is based on the 3 parameters like QoS, utilization of the resource, and energy. The solution matrix using minimum fitness value is selected to execute the task in their equivalent fitness order of VMs. The solution matrix is decided to exploit the arbitrarily selected solution to evaluate fitness. The solution matrix is decided from the arbitrarily chosen solution and it stands for the VM running the given task. The solution matrix is indicated as  $w$  and  $w_{jk}$  is the solution value of  $k^{\text{th}}$  task executed in  $j^{\text{th}}$  VM. From the solution matrix, the set matrix is decided which is attained by the division of the task length and the capability of the machine. The set matrix is indicated as  $S$ . Fitness evaluation is an important factor that decides the competent VM to run the task without distressing the system performance. In the task scheduling model, the fitness model is based upon 3 parameters, like the resource utilization, energy, and QoS. The fitness function returns the least value for an effectual solution. For the fitness function to stay at the least value, resource utilization ability of the VM is utmost, energy takes the least value, and the QoS takes the utmost value.

$$\text{Fitness } F = \frac{1}{3} \times [(1 - R_U) + E + (1 - \text{QoS})] \quad (5)$$

In eq. (12),  $E$  indicates the energy required to run a task in VM,  $R_U$  is the resource utilization, and QoS indicates the quality of service. The effectual scheduling of VM makes sure that the service is offered

to the users at a high-quality, inexpensive price, and competently. Moreover, the energy used by the VM in executing the task ought to stay at the least value, and the utilization resource is utmost. To emphasize that the utilization resource is utmost the term  $(1 - R_U)$  is augmented in the formula and the phrase  $(1 - QoS)$  denotes the utmost QoS.

### 5.1 QoS

The performance of the overall system is determined by the QoS, which is an important constraint. In a cloud computing environment, it is necessary to schedule VM with maximum quality. It primarily based upon 2 parameters, such as the time and the cost to execute the task in the VM. The cost and the time constraints are least to enable a better QoS to the users. The variables  $(1 - c)$  and  $(1 - t)$  indicates that the scheduling cost and time are least.

$$QoS = \frac{1}{2} [(1 - t) + (1 - c)] \quad (6)$$

In eq. (6),  $c$  indicates the execution cost of task execution and  $t$  indicates the execution time in the VM. The task execution time in the VM based upon the set matrix and the solution matrix. Eq. (7) denotes the execution time,  $S_{jk}$  indicates a set value of  $k^{th}$  task executed in  $j^{th}$  VM, and  $w_{jk}$  indicates the solution of the  $k^{th}$  task executed in the  $j^{th}$  VM.

$$t = \max_{\substack{\forall \\ k \in m}} [w_{jk} \times S_{jk}] \quad (7)$$

The execution cost of the VM is indicated in eq. (8) and (9),  $Y_{jk}$  indicate the product factor which is decided by  $S_{jk}$  and  $w_{jk}$ ,  $c$  indicates the cost of execution to execute tasks in  $j^{th}$  VM, and  $R_{jk}$  indicate the cost of resource in  $j^{th}$  VM. Hence, parameters which make sure utmost QoS are execution time, cost of execution, and the cost of the resource. These parameters have to have the least amount of value for the attainment of the utmost quality value that enhances the overall system performance.

$$c = \frac{1}{N} \sum_{j=1}^N Y_{jk} \times [R_{jk}] \quad (8)$$

and,

$$Y_{jk} = w_{jk} \times S_{jk} \quad (9)$$

### 5.2 Resource Utilization

The utilization of the resource for the VM is an additional factor that ought to be fundamentally utmost to improve the system performance. It is based upon set and the solution matrix and set and the solution value.

$$R_U = \frac{1}{m \times Q_1} \times \sum_{i=1}^m w_{jk} \times S_{jk} ; 1 \leq j \leq N ; 1 \leq k \leq m \quad (10)$$

$$\text{where, } Q_1 = \max(S) \quad (11)$$

In eq. (10),  $m$  indicates the total count of tasks,  $S$  indicates set matrix,  $w_{jk}$  indicates a solution of  $k^{th}$  task executed in the  $j^{th}$  VM and  $S_{jk}$  indicates the set value of  $k^{th}$  task executed in the  $j^{th}$  VM.  $N$  denotes the total number of VM in the cloud,  $Q_1$  parameter indicates decided based on the utmost set matrix value.

### 5.3 Energy

The VM energy to execute the task ought to be least for a capable model. The VM energy based upon the solution value of  $k^{th}$  task executed in  $j^{th}$  VM and the energy needed to execute  $k^{th}$  task executed in  $j^{th}$  VM.

$$E = \frac{1}{m \times Q_2} \sum_{k=1}^m w_{jk} \times \varepsilon_{jk} \quad (12)$$

$$Q_2 = \max(\varepsilon) \quad (13)$$

In eq. (12),  $m$  indicates the total count of tasks in VM, and  $Q_2$  indicates the factor which is the utmost of all the energy values necessary to task execution in the VM.

## 6. Multi-objective Task scheduling using the Proposed Method

### 6.1 Multi-Objective Function

The multi-objective task scheduling method exploiting the developed method is presented in this section. The tasks are scheduled to an exacting VM based on their execution cost, energy, and execution time. For competently task execution, the proposed method is exploited that obtains faster convergence to the optimal location with the minimum time and it saves the execution cost the task in the VM. The minimization of the cost and time of the execution enhances the effectiveness of the task scheduling method. Ahead of the arrival of the multiple user requests, the proposed method assigned the task to the additional competent VM and task execution to produce the optimized consequences.

### 6.2 Improved MSO Algorithm

In this paper, an enhanced model of the Moth Swarm Algorithm [7] named Improved MSA is developed based on a count of alterations on the construction of conventional MSA. These enhancements were used in transverse orientation and celestial navigation stages, as below:

a) In the transverse orientation stage, eq. (14) is changed as eq. (13) in the conventional MSA, to update the solution  $y_i^t$ . For that reason, the updated solution  $y_{i(new)}^{t+1}$  is computed using

Eq. (15):

$$y_i^{t+1} = |y_i^t - y_p^t \cdot PF| \cdot e^{\theta} \cdot \cos(2\pi\theta) + y_p^t \quad (14)$$

$$\forall p \in \{1, 2, \dots, n_p\}; i \in \{n_p + 1, n_p + 2, \dots, n_f\}$$

$$y_{i(new)}^{t+1} = y_{best}^t + \left[ cu_i \cdot |y_i^t - y_p^t \cdot PF| \cdot e^{\theta} \cdot \cos(2\pi\theta) \right] \quad (15)$$

$$\forall p \in \{1, 2, \dots, n_p\}; i \in \{n_p + 1, n_p + 2, \dots, n_f\}$$

In eq. (14), PF is the profit factor,  $y_{best}^t$  indicates the best solution;  $cu_i$  indicates a chaotic variable its value is produced using the Logistic map. The values of the profit factor are computed by eq. (16).

$$PF = \text{round}(1 + rn) \quad (16)$$

Eq. (15) has the possibility to search/use the solutions' space more effectual than eq. (14), for the reason that it permits the search both around the present solution  $y_i^t$  and around the optimal solution  $y_{best}^t$ .

b) The new onlooker moth in this sub-group  $y_i^{t+1}$  goes using a series of steps of Gaussian walks that are stated in eq. (17)

$$y_i^{t+1} = y_{best}^t + \varepsilon l + cu_i \cdot \left[ \varepsilon 2 \times best_g^t - \varepsilon 3 \times y_i^t \right] \forall i \in \{1, 2, \dots, n_G\} \quad (17)$$

In the celestial navigation stage, eq. (17) is changed as Eq. (18), to update the solution  $y_{i(new)}^{t+1}$ .

$$y_{i(new)}^{t+1} = y_{best}^t + \varepsilon l + cu_i \cdot \left[ \varepsilon 2 \times y_{best}^t - \varepsilon 3 \times y_i^t \right] \forall i \in \{1, 2, \dots, n_G\} \quad (18)$$

The parameters/variables have alike consequence using those stated in eq. (17) and (15). The individual formulations that compose relations (15) to (18) are modified from relations of symbiotic organism search approach [8].

c) In improved MSA, both estimate of the solution  $y_i^t$ , and its evaluation with one more solution (for instance,  $y_j^t$ ) is completed after updating/alteration of every module of the solution  $y_i^t$ . This approach assures that solution  $y_i^t$  keeps merely the modules that cause the development of the objective model  $F(y_i^t)$ . Similarly, for every alteration of a module from a solution  $y_i^t$ , it is verified if the optimal solution  $y_{best}^t$  is enhanced. Hence, if  $F(y_i^t) < F(y_{best}^t)$  subsequently the optimal solution  $y_{best}^t$  is updated with  $y_i^t (y_i^t \rightarrow y_{best}^t)$  and  $F(y_i^t) \rightarrow F(y_{best}^t)$  or else,  $y_{best}^t$  remnants unchanged.

d) In the conventional MSA approach, uniformly distributed arbitrary counts are produced in the range [0, 1] using the function  $\text{rand}[0,1]$ . Current studies stated that if chaotic maps are placed in the construction of a method, population diversity raises, hence capability of the method in exploring the searching space of solutions is enhanced and in conclusion, superior quality solutions will be attained. For that reason, in the developed method the chaotic series ( $cu_i = 1, 2, \dots, \infty$ ) produced with a Logistic map is represented rather than arbitrary numbers. Using the formulations, a chaotic sequence  $cu_i$  was produced by means of a Logistic map on every solution updation  $y_i^t$  in eq. (15) and (18).

The Logistic map is selected from the set of chaotic maps and it is stated in eq. (19).

$$cu_i = 4 \cdot cu_i(1 - cu_i); \quad cu_i \in (0,1) \quad (19)$$

In eq. (18),  $\{cu_i\}_i = 1, \dots, \infty$  indicates the numbers produced using a Logistic map, at  $i^{\text{th}}$  iteration. The preliminary circumstances are  $cu_0 \in (0,1)$ ; and  $cu_0 \neq \{0.0, 0.25, 0.5, 0.75, 1\}$ .

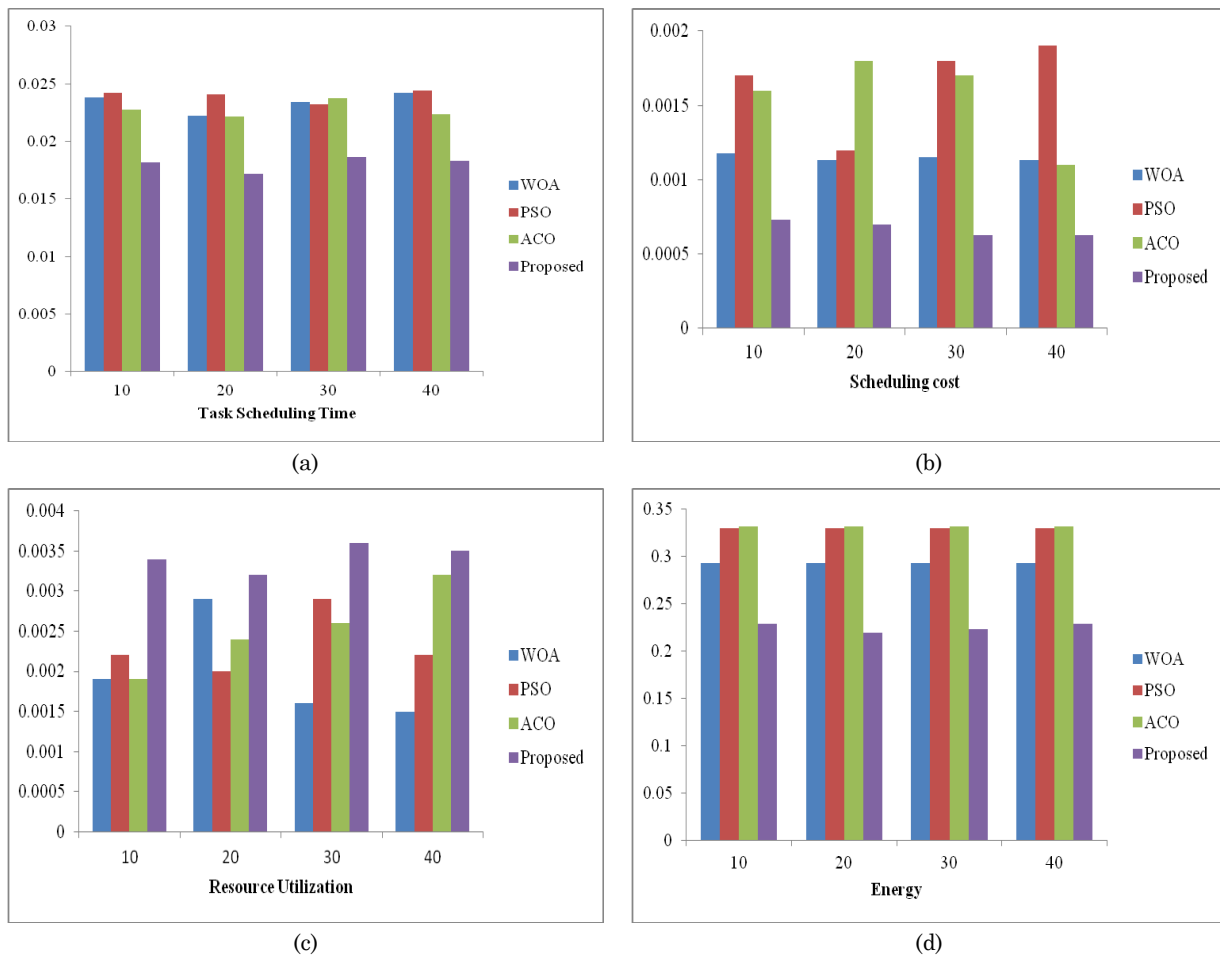
## 7. Results and Discussions

### 7.1 Experimental Procedure

The developed algorithm for scheduling the workflow and the outcomes of the developed algorithm were evaluated with the conventional techniques. The experimentation is carried out using the cloudsim tool with JAVA. The comparison is done regarding the evaluation metrics like the task scheduling time, the scheduling cost, energy, and resource utilization.

### 7.2 Performance Analysis

In Fig 2, the proposed algorithm attains a minimum value of the scheduling time and cost with the minimum energy. However, for conventional techniques like PSO, WOA, and ACO, the cost and scheduling time is higher than the developed technique. Concurrently, the resource utilization of the developed technique obtained the utmost value, which is higher while comparing with the other conventional techniques.



**Fig. 2.** Graphical representation of proposed and conventional models (a) Task scheduling time (b) scheduling cost (c) Resource utilization (d) Energy

## 5. Conclusion

This work exploits the developed model to perform the multi-objective task scheduling by exploiting a fitness parameter. The developed model was the enhanced version of the MOA method. Here, the optimal optimization solution based upon the fitness parameter and the fitness parameter value must remnant least to assure the utmost usage of the resources, least energy, and the between quality, and to attain customer fulfillment. The overall performance of the proposed method exhibits superior QoS and therefore, assures superior scheduling order to execute the tasks. The performance analysis was performed exploiting the experimental setups using the evaluation analysis of the developed model with conventional techniques. The analysis shows that the developed techniques attain a higher resource usage value and the least energy value, scheduling time, and cost at a rate correspondingly.

## References

- [1] Amrith Rajagopal Setlur, S. Jaya Nirmala, Har Simrat Singh, Sudhanshu Khoriya, "An efficient fault tolerant workflow scheduling approach using replication heuristics and checkpointing in the cloud", *Journal of Parallel and Distributed Computing*, Volume 13, 6February 2020, Pages 14-28.
- [2] Mutaz Barika, Saurabh Garg, Rajiv Ranjan, "Cost effective stream workflow scheduling to handle application structural changes", *Future Generation Computer Systems*, Volume 112, November 2020, Pages 348-361.
- [3] Zhaomeng Zhu, Xueyan Tang, "Deadline-constrained workflow scheduling in IaaS clouds with multi-resource packing", *Future Generation Computer Systems*, Volume 101, December 2019, Pages 880-893.
- [4] Goshgar Ismayilov, Haluk Rahmi Topcuoglu, "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing", *Future Generation Computer Systems*, Volume 102, January 2020, Pages 307-322.
- [5] Jin Sun, Lu Yin, Minhui Zou, Yi Zhang, Junlong Zhou, "Makespan-minimization workflow scheduling for complex networks with social groups in edge computing", *Journal of Systems Architecture*, Volume 108, September 2020.
- [6] Ying Xie, Yuanwei Zhu, Yeguo Wang, Yongliang Cheng, Yun Yang, "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud-edge environment", *Future Generation, Computer Systems*, Volume 97, August 2019, Pages 361-378.
- [7] Mohamed, A.A.A., Mohamed, Y.S., El-Gaafary, A.A., Hemeida, A.M., "Optimal power flow using moth swarm algorithm", *Elec. Power Syst. Res.* 142, 190e206, 2017
- [8] Akbarifard, S., Radmanesh, F., 2018. Predicting sea wave height using Symbiotic Organisms Search (SOS) algorithm. *Ocean Eng.* 167, 348e356.
- [9] VhatkarKapilNetaji, Bhole G P, "Optimal Container Resource Allocation Using Hybrid SA-MFO Algorithm in Cloud Architecture", *Multimedia Research*, Volume 3, Issue 1, January 2020.
- [10] Ashok Kumar C, Vimala R, "Load Balancing in Cloud Environment Exploiting Hybridization of Chicken Swarm and Enhanced Raven Roosting Optimization Algorithm", *Multimedia Research*, Volume 3, Issue 1, January 2020.
- [11] Santosh Kumar B. P, Venkata Ramanaiah K, "An Efficient Hybrid Optimization Algorithm for Image Compression", *Multimedia Research*, Volume 2, Issue 4, October 2019.