

# Gene Optimized Deep Neural Round Robin Workflow Scheduling in Cloud

Shanmugasundaram M<sup>1</sup>, Kittur H M<sup>3</sup>

School of Electronics Engineering  
Vellore Institute of Technology  
Vellore, India

Kumar R<sup>2</sup>

Department of Electronics and Instrumentation Engineering  
National Institute of Technology  
Nagaland

**Abstract**—Workflow scheduling is a key problem to be solved in the cloud to increase the quality of services. Few research works have been designed for performing workflow scheduling using different techniques. But, scheduling performance of existing techniques was not effective when considering a larger number of user tasks. Besides, the makespan of workflow scheduling was higher. In order to solve such limitations, Gene Optimized Deep Neural Round Robin Scheduling (GODNRRS) Technique is proposed. The designed GODNRRS Technique contains three layers namely input, hidden and output layer to efficiently perform workflow scheduling in the cloud. The GODNRRS Technique initially gets the number of user tasks as input in the input layer and forwards it to the hidden layer. After taking input, GODNRRS Technique initializes gene population with the assist of virtual machines in Amazon cloud server at the first hidden layer. Next, GODNRRS Technique determines fitness function for each virtual machine using their energy, memory, CPU time, bandwidth capacity at the second hidden layer. Afterward, GODNRRS Technique defines a weight for each virtual machine at the third hidden layer depends on their fitness function estimation. Consequently, GODNRRS Technique distributes the user tasks to optimal virtual machines according to their weight value at the fourth hidden layer in a cyclic manner. At last, the output layer renders the scheduled tasks result. Thus, GODNRRS Technique handles workflows in the cloud with improved scheduling efficiency and lower energy and makespan. The GODNRRS Technique conduct the experimental evaluation using metrics such as scheduling efficiency, makespan, and energy consumption with respect to a different number of user tasks from LIGO, Montage and cybershake real-time applications. The experimental result show that the GODNRRS Technique is able to increase the efficiency and also reduces the makespan of workflows scheduling in the cloud as compared to state-of-the-art works.

**Keywords**—Bandwidth capacity; processor time; energy; fitness function; memory; user task; virtual machine

## I. INTRODUCTION

Cloud computing is promising with increasing popularity in workflow scheduling. Workflow scheduling plays an imperative part in cloud computing where it assigns tasks to suitable resources to execute. Effective task scheduling is required for obtaining high performance in a cloud environment. Many workflows scheduling method has been designed in the existing works. But, scheduling performance of conventional techniques was not efficient. Therefore,

GODNRRS Technique is proposed in this work to increase the workflow scheduling efficiency in the cloud.

Genetic-based algorithm (GA) was designed in [1] with aiming at optimizing the workflow scheduling performance in the cloud. However, scheduling efficiency was poor. A dynamic cost-effective deadline-constrained heuristic algorithm was introduced in [2] for scheduling a scientific workflow in a public cloud. But, the makespan of scheduling was higher.

A Dynamic Priority-Based Approach was presented in [3] for efficient workflow scheduling in a cloud computing system. However, handling multiple workflows and dynamic workflow scheduling was not considered. Particle swarm optimization (PSO) was intended in [4] to reduce the overall workflow execution cost while meeting deadline constraints. But, the energy of the virtual machine was not considered in this method.

The chemical reaction optimization (CRO) and ant colony optimization (ACO) algorithms were combined in [5] to resolve the workflow-scheduling problem in Cloud. However, scheduling performance was lower. Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows was presented in [6] to reduce the execution time and economic cost. But, energy utilized by the virtual machine was higher.

Multi-objective workflow scheduling was introduced in [7] with the application of a heuristic algorithm depends on the task's completion time. However, the false positive rate of workflow scheduling was more. An evolutionary multi-objective optimization (EMO)-based algorithm was developed in [8] in order to address the workflow scheduling issues in infrastructure as a service (IaaS) platform. But, the time complexity of scheduling was more.

Pareto-based multi-objective workflow scheduling algorithm was designed in [9] in order to lessen makespan and to increase energy efficiency in the cloud. However, workflow-scheduling was not improved. A multi-swarm multi-objective optimization algorithm (MSMOOA) was designed in [10] to perform workflow scheduling in the cloud. But, energy and makespan of scheduling remained an open issue.

In order to address the above mentioned existing drawbacks in cloud workflow scheduling, GODNRRS Technique is developed in this research work. The main contributions of GODNRRS technique are described in below.

- To increase the workflow scheduling performance in a cloud environment with a minimal makespan when compared to state-of-the-art works, GODNRRS technique is proposed. By using the deep neural learning, GODNRRS Technique handles the larger number of workflows in the cloud with lower time complexity. Further, round-robin scheduling helps for GODNRRS Technique distributes a number of user tasks to optimal virtual machines in a cycle manner.
- To reduces the energy usage in the cloud as compared to conventional works, a genetic algorithm is applied in GODNRRS technique where it finds optimal virtual machines using fitness evaluation (i.e. energy, memory, CPU time, bandwidth capacity) to carry out user tasks on a cloud server.

The rest of the paper structure is formulated as follows: Section II reviews the related works. In Section III, the proposed GODNRRS Technique is explained with assists of architecture diagram. In Section IV, Simulation settings are described and the result discussion is presented in Section V. Section VI presents the conclusion of the paper.

## II. RELATED WORKS

Budget Deadline Aware Scheduling (BDAS) was introduced in [11] to increase the task parallelism through partitioning tasks. A service-oriented scheduler planner was developed in [12] to get resource efficiency, overall controlled performance and task prioritization for continuous data processing workflows.

A cloud workflow scheduling problem was solved in [13] with the help of a hybrid resource provisioning method to lessen the total renting cost. A Completion Time Driven Hyper-Heuristic (CTDHH) approach was presented in [14] for cost optimization in Scientific Workflow Applications.

A novel scheduling algorithm was designed in [15] for cloud computing using the driver of the dynamic essential path (DDEP) by considering actual computation cost and communication cost of task node in the scheduling process. A Hybrid GA-PSO algorithm was intended in [16] to decrease the makespan and the cost and balance the load of the dependent tasks over the heterogonous resources in the cloud.

A workflow task scheduling algorithm based on the resources fuzzy clustering was introduced in [17] to minimize makespan of the precedence constrained applications. Cost-Time Efficient Scheduling Plan was implemented in [18] for executing workflows in the cloud.

A Deadline-constrained Cost Optimization Approaches was designed in [19] with the application of ant colony optimization for workflow scheduling in clouds. A fuzzy dominance sort based heterogeneous earliest-finish-time (FDHEFT) algorithm was presented in [20] to diminish cost and makespan for workflow scheduling in the cloud. An improved round robin (IRR) algorithm was presented in [21] for workflow scheduling in cloud.

## III. GENE OPTIMIZED DEEP NEURAL NETWORK ROUND ROBIN SCHEDULING TECHNIQUE

Cloud computing technology uses the internet to render scalable services for its users. Besides, cloud employs a huge amount of heterogeneously distributed resources to provide countless different services to its users with quality of service (QoS) requirements. Workflow scheduling is one of the considerable problems to be resolved in cloud computing. A lot of workflow scheduling schemes has been intended in the existing works. However, scheduling performance of conventional techniques was not effectual. Furthermore, the makespan of workflow scheduling was more. In order to overcome the above drawbacks, Gene Optimized Deep Neural Round Robin Scheduling (GODNRRS) Technique is developed. The GODNRRS Technique improves the workflow scheduling performance in a cloud environment. The GODNRRS Technique is designed to reduce the execution time of workflow and reduce energy utilization. The GODNRRS Technique is proposed by combining the genetic algorithm, round-robin scheduling in deep neural learning on the contrary to state-of-the-art works.

The deep neural learning is used in GODNRRS Technique on the contrary to existing scheduling algorithm as it helps for efficient workflow scheduling with minimal time when the number of user tasks to the cloud server is more. Deep neural learning depends on artificial neural networks (ANNs). Besides to that, a genetic algorithm is employed in GODNRRS Technique as it assists for finding optimal virtual machines based on fitness evaluation (i.e. energy, memory, CPU time, bandwidth capacity) to perform tasks. Moreover, round-robin scheduling is applied in GODNRRS Technique to schedule the user tasks to optimal virtual machines in a cycle manner based on their weight value. This supports for GODNRRS Technique to significantly carry out workflow scheduling process in the cloud with higher efficiency. Thus, GODNRRS Technique also balances the workloads of virtual machines in the cloud. The architecture diagram of GODNRRS technique is depicted in Fig. 1.

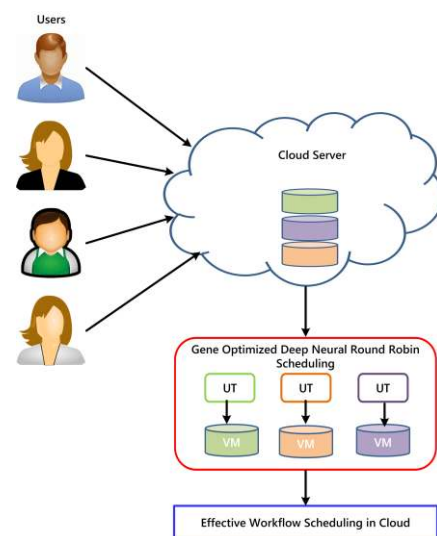


Fig. 1. Architecture of GODNRRS Technique for Workflow Scheduling.

Fig. 1 presents the overall processes of GODNRRS technique to attain improved efficiency for scheduling the workflows in the cloud. Let us consider a Cybershake, LIGO and Montage real-time applications where the users request numerous numbers of tasks to the cloud server.

As demonstrated in the above figure, GODNRRS Technique takes a number of user tasks ‘ $UT_i = UT_1, UT_2, UT_3 \dots UT_n$ ’ from Cybershake, LIGO and Montage real time applications as input. Then, Gene Optimized Deep Neural Round Robin Scheduling is applied with the aim of scheduling the tasks to best virtual machines ‘ $VM_i = VM_1, VM_2, VM_3, \dots, VM_n$ ’ and thereby minimizing the makespan and energy in cloud. The detailed processes of GODNRRS Technique are described in below.

The GODNRRS technique is an artificial neural network (ANN) with many layers between the input and output layers. The GODNRRS Technique is a feed-forward network where data flows from the input layer to the output layer without looping back. Fig. 2 demonstrates the structure of GODNRRS Technique.

Fig. 2 shows the flow processes of GODNRRS Technique to manage the workflows on a cloud server. As presented in the below figure, the GODNRRS structure contains three layers such as input, hidden, output layer. These three layers are interconnected to each other. The first layer contains input neurons that get a number of incoming user tasks to the cloud server as input and transmit taken input to the second layer known as the hidden layer. In GODNRRS Technique, four hidden layers are used where the initial population, fitness evaluation, weight determination, and round-robin scheduling is carried out to assign tasks to optimal virtual machines (VMs) and thereby managing the workflows among diverse virtual machines. At last, the output layer returns the scheduled user tasks result with higher efficiency.

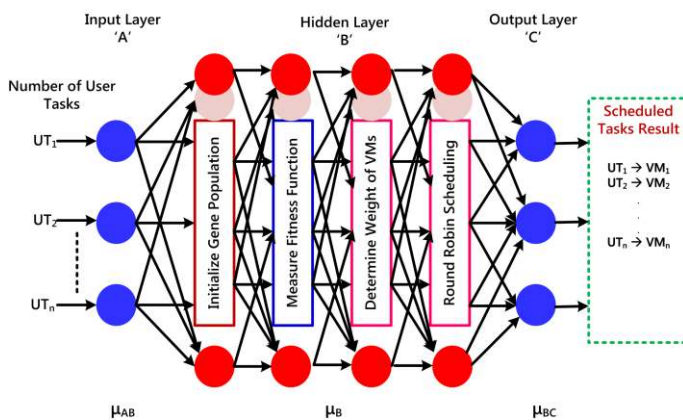


Fig. 2. Structure of GODNRRS Technique.

Let us consider a number of user tasks are ‘ $UT_1, UT_2, UT_3, \dots, UT_n$ ’ to cloud server (CS). This cloud server renders multiple services to users by assigning the tasks to virtual machines. Let us assume a cloud server includes of number of virtual machines represented as ‘ $VM_1, VM_2, VM_3, \dots, VM_n$ ’ . The GODNRRS technique acquires user tasks ‘ $UT_i$ ’ as input. After obtaining the input, the input layer ‘A’ in GODNRRS technique unites the user tasks ‘ $UT_i$ ’ with weights  $\mu_{AB}$  and bias

term  $x_j$ . Hence, the neurons process in the input layer ‘(A)’ obtained mathematically as,

$$A(t) = \sum UT_i \mu_{AB} + x_j \tag{1}$$

From equation (1), ‘ $A(t)$ ’ refers action of a neuron in input layer at a time ‘ $t$ ’. Here, ‘ $UT_i$ ’ denotes a user tasks to amazon cloud server and ‘ $\mu_{AB}$ ’ represents the weight between the input and hidden layer and ‘ $x_j$ ’ is a bias term. After obtaining the input, the input layer forwards the user tasks to hidden layer. The first hidden layers in GODNRRS Technique initialize the gene population with help of diverse number of virtual machines in cloud server. Therefore, neurons action in first hidden layer ‘ $B_1(t)$ ’ is determined mathematically as,

$$B_1(t) = \sum A(t) \mu_{B1}, IGP_{VM_i} \tag{2}$$

From equation (2), ‘ $B_1(t)$ ’ point outs the outcome of first hidden layer at time ‘ $t$ ’ whereas ‘ $A(t)$ ’ indicates the input get from the input layer (i.e. number of user tasks ‘ $UT_i = UT_1, UT_2, UT_3 \dots UT_n$ ’ and ‘ $\mu_{B1}$ ’ refers weight of first hidden layers. Here, ‘ $IGP_{VM_i}$ ’ represents the initial population of genes with virtual machines for an incoming user tasks at the time which is mathematically formulated as:

$$IGP_{VM_i} = VM_1, VM_2, \dots, \dots, VM_n \tag{3}$$

From (3), ‘ $n$ ’ designates a number of virtual machines in cloud server. After completing initialization process, GODNRRS technique determines the fitness function for each virtual machine based on energy, memory, CPU time, bandwidth capacity. Thus, neurons action in second hidden layer ‘ $B_2(t)$ ’ at a time ‘ $t$ ’ is mathematically expressed as,

$$B_2(t) = \sum B_1(t) \mu_{B2}, F_{VM_i} \tag{4}$$

From equation (4), ‘ $B_2(t)$ ’ signify the result of second hidden layer and ‘ $B_1(t)$ ’ denotes the input obtained from the first hidden layer i.e. initial population of genes with VMs. Here ‘ $\mu_{B2}$ ’ is weight of second hidden layers and ‘ $F_{VM_i}$ ’ point outs the fitness function of each virtual machine in cloud server. From that, fitness function of VMs is mathematically calculated as:

$$F_{VM_i} = [e_{VM_i}, m_{VM_i}, c_{VM_i}, b_{VM_i}] \tag{5}$$

From equation (5), fitness function of each virtual machine ‘ $F_{VM_i}$ ’ is measured by considering energy ‘ $(e_{VM_i})$ ’, memory ‘ $(m_{VM_i})$ ’, CPU time ‘ $(c_{VM_i})$ ’, bandwidth ‘ $(b_{VM_i})$ ’ capacity. In GODNRRS Technique, energy estimates an amount of remaining power available in virtual machine to perform arrival tasks and to provide the required services to user in cloud. Accordingly, energy is determined as difference between the total energy ‘ $e_T$ ’ and the energy used ‘ $e_U$ ’ by virtual machine using below,

$$e_{VM_i} = e_T - e_U \tag{6}$$

From equation (6), the residual energy capacity of each virtual machine ‘ $e_{VM_i}$ ’ is determined. The energy is estimated in terms of a joule (J). Followed by, memory calculates the amount of storage space available in virtual machine to carry out the user tasks. From that, memory is measured as

differentiation between the total memory ‘ $m_T$ ’ and the memory utilized ‘ $m_U$ ’ by virtual machine using below,

$$m_{VM_i} = m_T - m_U \quad (7)$$

From equation (7), the remaining memory capacity of all virtual machine ‘ $m_{VM_i}$ ’ is obtained. Then, CPU time is measured as difference between total CPU time ‘ $c_T$ ’ and the amount of CPU time utilized by users ‘ $c_U$ ’ using below:

$$c_{VM_i} = c_T - c_U \quad (8)$$

From equation (8), the residual CPU capacity of all virtual machine ‘ $c_{VM_i}$ ’ is determined. Afterward, Bandwidth is estimated as differentiation between the total bandwidth ‘ $b_T$ ’ and bandwidth utilized by virtual machine ‘ $b_U$ ’ using below,

$$b_{VM_i} = b_T - b_U \quad (9)$$

From equation (9), the residual bandwidth capacity of each virtual machine ‘ $b_{VM_i}$ ’ is obtained. Based on the estimated fitness functions (i.e. available energy, memory, CPU time, bandwidth capacity), third hidden layer in GODNRRS Technique assigns weight to each virtual machine. Hence, neurons process in third hidden layer ‘ $B_3(t)$ ’ is obtained mathematically as,

$$B_3(t) = \sum B_2(t) \mu_{B_3}, W_i (VM_i) \quad (10)$$

From equation (10), ‘ $B_3(t)$ ’ is a result of third hidden layer at a time ‘ $t$ ’. Here, ‘ $B_2(t)$ ’ indicates the input get from the second hidden layer (i.e. determined fitness function) and Here, ‘ $\mu_{B_3}$ ’ point outs weight of third hidden layers and ‘ $W_i (VM_i)$ ’ refers the weight value of virtual machine. In GODNRRS Technique, weight is determined for each virtual machine in cloud server by considering fitness function where weight is an integer value. The GODNRRS Technique assigns weight value based on the residual processing capacity of virtual machine i.e. energy, memory, CPU time, and bandwidth capacity. Subsequently, weight of virtual machine is computed mathematically as:

$$W_i(VM_i) \rightarrow \sum_{i=1}^n F_{VM_i} \quad (11)$$

From equation (11), the weight value is discovered for each virtual machine. Thus, GODNRRS Technique schedules the user tasks first to a virtual machine with higher weight than those with a minimal weight value. In the fourth hidden layer, GODNRRS Technique accomplishes round-robin scheduling where user tasks are allocated to an optimal virtual machine according to their weight value. As a result, neurons activity in fourth hidden layer ‘ $B_4(t)$ ’ is mathematically determined as,

$$B_4(t) = \sum B_3(t) \mu_{B_4}, RRS (UT_i \rightarrow VM_i) \quad (12)$$

From equation (12), ‘ $B_4(t)$ ’ indicates the fourth hidden layer output at a time ‘ $t$ ’ and ‘ $B_3(t)$ ’ is the input taken from the three hidden layer (i.e. weight value of virtual machine) and ‘ $\mu_{B_4}$ ’ represents weight of fourth hidden layers. Here, ‘ $RRS (UT_i \rightarrow VM_i)$ ’ denotes the scheduling of user tasks to optimal virtual machines according to their weight values in round robin manner. For instance, if weight of ‘ $VM_1$ ’ is 5 and ‘ $VM_2$ ’ is 8, the GODNRRS Technique schedule 3 user tasks to ‘ $VM_1$ ’ and 5 user tasks to ‘ $VM_2$ ’ in a cycle manner.

This supports for GODNRRS Technique to effectively distribute the user tasks to the best virtual machines in cloud server as compared to state-of-the-art works. The scheduled tasks result is then sent to output layer. The process in output layer ‘ $C(t)$ ’ at time ‘ $t$ ’ is formulated mathematically as:

$$C(t) = Y \mu_{BC} B_4(t) \quad (13)$$

From equation (13), ‘ $\mu_{BC}$ ’ is a weight between the hidden and output layer and ‘ $Y$ ’ is an activation function. The output layer in GODNRRS Technique gives scheduled result of tasks to optimal virtual machines. By using the above processes, the GODNRRS Technique efficiently handle the workflows in cloud with minimal energy and makespan.

The algorithmic process of GODNRRS technique is explained in below,

---

**Algorithm: GODNRRS Algorithm**

---

<b>Input:</b>	Number of cloud tasks ‘ $UT_1, UT_2, UT_3, \dots, UT_n$ ’; Cloud server ‘ $CS$ ’; Number Of Virtual Machines ‘ $VM_1, VM_2, VM_3, \dots, VM_n$ ’
<b>Output:</b>	Improved efficiency for workflow scheduling in cloud
<b>Step 1:</b>	<b>Begin</b>
<b>Step 2:</b>	For ‘ $UT_i \in ACS$ ’
<b>Step 3:</b>	Takes a number of tasks as input using (1) // <b>Input Layer</b>
<b>Step 4:</b>	Forwards incoming ‘ $UT_i$ ’ to hidden layer
<b>Step 5:</b>	Initialize gene populations with VMs using (2), (3) // <b>First Hidden Layer</b>
<b>Step 6:</b>	<b>For each ‘<math>VM_i</math>’ // Second Hidden Layer</b>
<b>Step 7:</b>	Measure ‘ $e_{VM_i}$ ’, ‘ $m_{VM_i}$ ’, ‘ $c_{VM_i}$ ’, ‘ $b_{VM_i}$ ’ using (6), (7), (8), (9)
<b>Step 8:</b>	Calculate ‘ $F_{VM}$ ’ using (4) and (5)
<b>Step 9:</b>	<b>End for</b>
<b>Step 10:</b>	Determine weight of each ‘ $F_{VM}$ ’ using (10), (11) // <b>Third Hidden Layer</b>
<b>Step 11:</b>	Schedule ‘ $UT_i$ ’ to optimal VM based on weight using (12) // <b>Fourth hidden layer</b>
<b>Step 12:</b>	Returns scheduled tasks result using (13) // <b>Output layer</b>
<b>Step 13:</b>	<b>Endif</b>
<b>Step 14:</b>	<b>End</b>

---

The above algorithm describes the step by step process of GODNRRS technique to get better workflow scheduling performance in the cloud. As shown in the above algorithmic process, the number of tasks to the cloud server is obtained as input in the input layer. Then, the gene population is initialized with the help of a number of virtual machines at the first hidden layer. After that, fitness function is evaluated for each virtual machine depends on energy, memory, CPU time, bandwidth capacity at the second hidden layer. Followed by, the weight value is defined for each virtual machine by considering their fitness function at the third hidden layer. Subsequently, assignment of user tasks to best virtual machines is carried in a cyclic way based on their weight value at the fourth hidden layer. Finally, the output layer provides the scheduled tasks result. By using the above algorithmic process, GODNRRS Technique efficiently handles workflows in the

cloud with higher efficiency and minimal energy and makespan as compared to existing works.

#### IV. EXPERIMENTAL SETUP

In order to evaluate the performance of proposed, GODNRRS Technique is implemented in Eclipse IDE Java Language in Cloudsim simulator using Cybershake, LIGO and Montage real-time applications. The GODNRRS Technique considers the various number of user tasks in the range of 100-1000 from LIGO and Montage real-time applications for conducting the experimental process. These the workflows are run on Amazon's cloud computing platform EC2 to get required services. Montage is an astronomical application which is employed to create custom mosaics of the sky based on a set of images. Most of its tasks are represented as I/O intensive which does not need much processing capacity. Besides, LIGO workflow is utilized in gravitational physics for identifying gravitational waves generated by different events in the universe. This workflow is represented as having CPU intensive tasks. The Cybershake is employed in earthquake science to indicate earthquake hazards in a region through generating synthetic seismograms. The Amazon EC2 comprises of attributes such as Name, API Name, Memory, Compute Units (ECU), Cores, Storage, Arch, Network Performance, Max Bandwidth (MB/s), Max IPs, Linux cost and Windows cost. With the help of the Cloudsim simulator, GODNRRS technique renders demanded services to cloud users with the available resources through efficient workflow scheduling. The system configuration is shown in below Table I.

TABLE I. SYSTEM CONFIGURATION

Parameters	Quantity
Number of users	1
Number of virtual machine	20
Number of Data centers	1
RAM size	2 GB
Processor Speed	3.40 GHz
Number of processors	1
Bandwidth	20 MB
Architecture	X86

The performance of GODNRRS technique is estimated in terms of scheduling efficiency, makespan, and energy consumption. The experimental result of proposed technique is compared against two conventional methods namely Multi-Population Genetic Algorithm (MPGA) [1] and Just-in-time (JIT) Workflow Scheduling [2] and Improved Round Robin (IRR) [21].

#### V. RESULTS AND DISCUSSIONS

In this section, the performance result of GODNRRS technique is discussed. The simulation result of proposed technique is compared with Multi-Population Genetic Algorithm (MPGA) and Just-in-time (JIT) Workflow Scheduling and Improved Round Robin (IRR) respectively using below parameters with the assist of tables and graphs.

##### A. Scheduling Efficiency

Scheduling efficiency '(SE)' evaluates the ratio of number of user tasks that correctly scheduled to the best virtual machine to the total number of user tasks. The scheduling efficiency is mathematically formulated as,

$$SE = \frac{CS_{UR_i}}{N} * 100 \quad (14)$$

From equation (14), 'N' point outs a number of user tasks and 'CS<sub>UR<sub>i</sub></sub>' refers the number of user task that accurately scheduled. The scheduling efficiency is determined in percentage (%).

Sample Mathematical Calculation for Scheduling Efficiency using LIGO Workflow:

- Existing MPGA: the number of user tasks exactly scheduled is 65 and total number of user tasks is 100. Then, scheduling efficiency is obtained as,

$$SE = \frac{88}{100} * 100 = 88\%$$

- Existing JIT Workflow Scheduling: the number of user tasks precisely scheduled is 70 and total number of user tasks is 100. Then, scheduling efficiency is determined as,

$$SE = \frac{89}{100} * 100 = 89\%$$

- Existing IRR: the number of user tasks accurately scheduled is 79 and total number of user tasks is 100. Then, scheduling efficiency is determined as,

$$SE = \frac{92}{100} * 100 = 92\%$$

- Proposed GODNRRS Technique: the number of user tasks correctly scheduled is 90 and total number of user tasks is 100. Then, scheduling efficiency is measured as,

$$SE = \frac{99}{100} * 100 = 99\%$$

The GODNRRS technique is implemented in Eclipse IDE Java Language using Cloudsim simulator by considering the real application workflows namely Cybershake, LIGO and Montage in order to estimate the scheduling efficiency. The GODNRRS Technique takes a different number of user tasks in the range of 100-1000 from above real-time application to conduct experimental process. When performing the experimental evaluation using 500 user tasks from Montage real-time application, GODNRRS Technique gets 96 % scheduling efficiency whereas existing MPGA, JIT Workflow Scheduling and IRR obtains 84%, 86 % and 86 % respectively. From that, it is significant that the scheduling efficiency using proposed GODNRRS technique is higher than other conventional works. The tabulation result analysis of scheduling efficiency for LIGO and Montage, cybershake workflows is presented in below Table II.

Fig. 3(a, b, c) demonstrates the experimental results of scheduling efficiency for Cybershake, LIGO and Montage real-time applications with respect to a dissimilar number of user tasks using three methods namely MPGA, JIT Workflow Scheduling, IRR and GODNRRS technique. As presented in Fig. 3, proposed GODNRRS technique achieves higher scheduling efficiency for handling workflows in Cybershake, LIGO and Montage real-time applications when compared to MPGA, JIT Workflow Scheduling and IRR. This is due to the integration of genetic algorithm, round-robin scheduling and deep neural learning in proposed GODNRRS technique on the contrary to state-of-the-art works. By using the algorithmic processes of GODNRRS, proposed technique increases the workflows scheduling performance in the cloud for the above real-time applications by assigning tasks to optimal VMs based on their weight. This supports for proposed technique to increase the ratio of number of user tasks that correctly scheduled to the optimal virtual machines as compared to existing works. Hence, the proposed GODNRRS technique improves the workflow scheduling efficiency of LIGO application by 16%, 12.5% and 8 % as compared to compared to MPGA, JIT Workflow Scheduling and IRR respectively. In the same way, proposed GODNRRS technique enhances the workflow scheduling efficiency of Montage application by 19%, 13 % and 9 % as compared to compared to MPGA, JIT Workflow Scheduling and IRR respectively. Likewise, this technique increases the workflow scheduling efficiency of Cybershake application by 18%, 14 % and 8 % as compared to compared to MPGA, JIT Workflow Scheduling and IRR respectively.

**B. Makespan**

The makespan ‘M’ represents the total length of the schedule. Thus, the makespan determines the amount of time

taken to schedule all user tasks to the optimal virtual machines. The makespan is mathematically computed using below,

$$M = N * t(SS_{UT}) \tag{15}$$

From equation (15), ‘N’ indicates a number of user tasks considered for experimental work and ‘t(SS<sub>UT</sub>)’ denotes the time employed for scheduling the single user tasks. The makespan is estimated in milliseconds (ms).

Sample Mathematical Calculation for Makespan using LIGO Workflow:

- Existing MPGA: the time employed to schedule the single user task is 0.35 ms and total number of tasks is 100. Then, makespan is estimated as,

$$M = 100 * 0.35 = 35ms$$

- Existing MPGA: the amount of time utilized to schedule the single user task is 0.33 ms and total number of user tasks is 100. Then, makespan is calculated as,

$$M = 100 * 0.33 = 33ms$$

- Existing IRR: the amount of time used to schedule the single user task is 0.30 ms and total number of user tasks is 100. Then, makespan is obtained as,

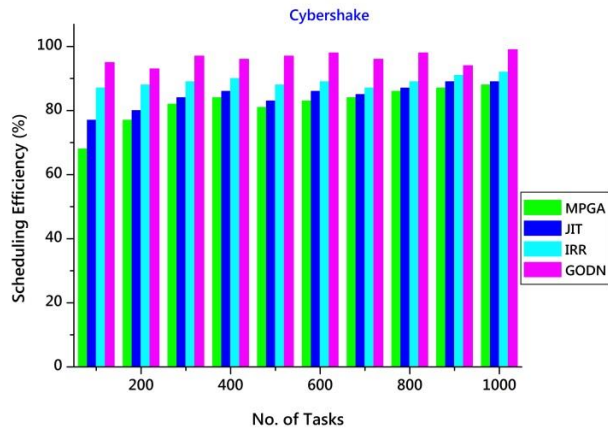
$$M = 100 * 0.30 = 30ms$$

- Proposed GODNRRS Technique: the amount of time needed to schedule the single user task is 0.28 ms and total number of user tasks is 100. Then, makespan is determined as,

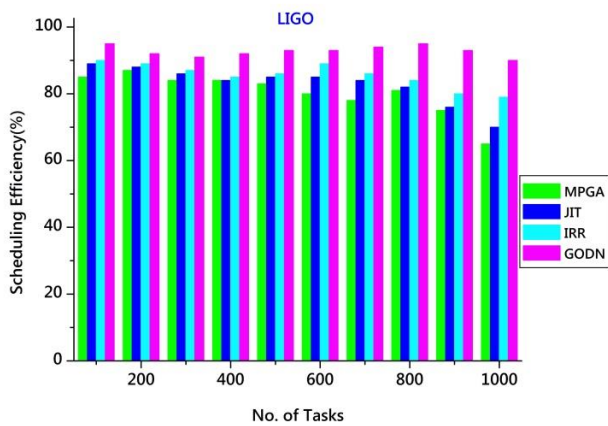
$$M = 100 * 0.28 = 28ms$$

TABLE II. TABULATION FOR SCHEDULING EFFICIENCY

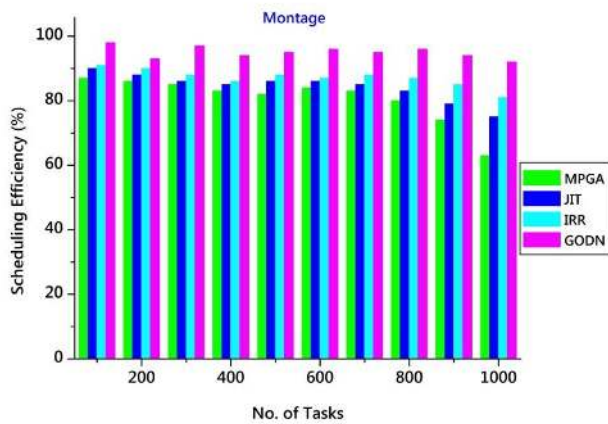
Number of user tasks (N)	Scheduling Efficiency (%)											
	Cybershake Workflow				LIGO Workflow				Montage Workflow			
	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS
100	88	89	92	99	85	89	90	95	87	90	91	98
200	87	89	91	94	87	88	89	92	86	88	90	93
300	86	87	89	98	84	86	87	91	85	86	88	97
400	84	85	87	96	84	84	85	92	83	85	86	94
500	83	86	89	98	83	85	86	93	82	86	88	95
600	81	83	88	97	80	85	89	93	84	86	87	96
700	84	86	90	96	78	84	86	94	83	85	88	95
800	82	84	89	97	81	82	84	95	80	83	87	96
900	87	89	91	94	87	88	89	92	86	88	90	93
1000	88	89	92	99	85	89	90	95	87	90	91	98



(a) Scheduling Efficiency Analysis for Cybershake Workflow.



(b) Scheduling Efficiency Analysis for LIGO Workflow.



(c) Scheduling Efficiency Analysis for Montage Workflow.

To determine the makespan involved during processes of cloud workflow scheduling, the GODNRRS technique is implemented in Eclipse IDE Java Language using Cloudsim

simulator with two real application workflows namely cybershake, LIGO and Montage. This technique considers a varied number of user tasks in the range of 100-1000 from cybershake, LIGO and Montage real-time application to accomplish experimental work. When taking 700 user tasks from LIGO application to accomplish the experimental process, the proposed technique acquires 70 ms makespan whereas conventional works MPGA and JIT Workflow Scheduling and IRR gets 105 ms, 98 ms and 84 ms respectively.

From that it is important that the makespan using proposed GODNRRS technique is lower than other state-of-the-art works. The comparative result analysis of makespan for LIGO and Montage, cybershake workflows is depicted in below Table III.

Fig. 4(a, b, c) shows the experimental results of Makespan for both LIGO and Montage real-time applications versus a various number of user tasks using three methods namely MPGA, JIT Workflow Scheduling, IRR and GODNRRS technique. As depicted in Fig. 4, proposed GODNRRS technique attains minimal Makespan to schedule all workflows in Cybershake, LIGO and Montage real-time applications when compared to MPGA, JIT Workflow Scheduling and IRR. This is owing to a combination of genetic algorithm, round-robin scheduling and deep neural learning in proposed technique on the contrary to conventional works. With the support of the algorithmic steps of GODNRRS, proposed technique schedules all tasks in Cybershake, LIGO and Montage real-time applications to best VMs according to their weight value with a minimal amount of time. This helps for proposed GODNRRS technique to minimize the amount of time taken to effectively schedule all user tasks in the considered real-time applications as compared to existing works. Thus, the proposed technique decreases the Makespan of LIGO workflow scheduling by 28%, 22 % and 11 % when compared to MPGA, JIT Workflow Scheduling, and IRR respectively. Likewise, it also reduces the Makespan of Montage workflow scheduling by 46%, 33 % and 25 % as compared to compared to MPGA, JIT Workflow Scheduling, and IRR respectively. Similarly, the proposed GODNRRS technique minimizes the Makespan of Cybershake workflow scheduling by 45%, 43 % and 22 % as compared to compared to MPGA, JIT Workflow Scheduling, and IRR respectively.

### C. Energy Consumption

Energy consumption 'EC' calculates an amount of energy utilized by the virtual machine to carry out user task. The energy consumption is calculated using below,

$$EC = N * e(S_{UT_i}) \quad (16)$$

From equation (16), 'e(S<sub>UT<sub>i</sub>)' is the energy used to accomplish the single user task and 'N' refers to a number of user tasks considered during the experimental process. The energy consumption is estimated in terms of a joule '(J)' .</sub>

TABLE III. TABULATION FOR MAKESPAN

Number of user tasks (N)	Makespan (ms)											
	Cybershake Workflow				LIGO Workflow				Montage Workflow			
	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS
100	34	32	27	23	35	33	30	28	36	31	29	25
200	46	42	32	26	48	44	38	34	50	40	36	28
300	66	60	45	36	69	63	54	48	72	57	51	39
400	76	72	52	40	80	76	64	56	84	68	60	44
500	85	80	55	40	90	85	70	60	95	75	65	45
600	96	90	60	42	102	96	78	66	108	84	72	48
700	98	91	63	46	105	98	84	70	112	84	77	49
800	112	104	72	56	120	112	96	88	128	96	88	64
900	117	108	77	59	126	108	99	90	135	99	90	63
1000	100	120	90	82	110	100	97	95	120	110	100	92

Sample Mathematical Calculation for Energy Consumption using LIGO Workflow:

- Existing MPGA: the amount of energy required to perform a single user task is 0.54 J and the total number of user tasks is 100. Then energy consumption is determined as,

$$EC = 100 * 0.54 = 54J$$

- Existing JIT Workflow Scheduling: the amount of energy desired to complete a single user task is 0.53 J and the total number of user requests is 100. Then energy consumption is calculated as,

$$EC = 100 * 0.53 = 53J$$

- Existing IRR: the amount of energy used to complete a single user task is 0.51 J and the total number of user requests is 100. Then energy consumption is formulated as,

$$EC = 100 * 0.51 = 51J$$

- Proposed GODNRRS Technique: the amount of energy utilized to carry out a single user task is 0.46 J and the total number of user tasks is 100. Then energy consumption is measured as,

$$EC = 100 * 0.46 = 46J$$

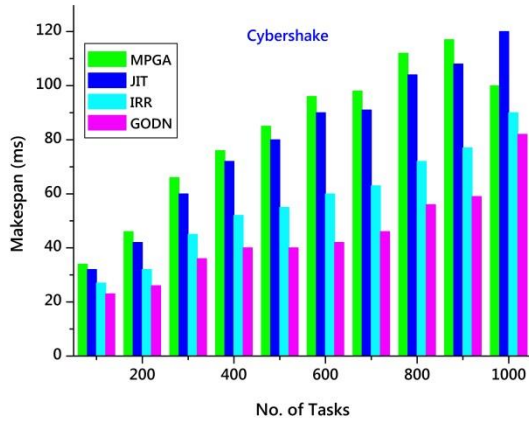
In order to calculate the amount of energy taken for accomplishing the user tasks in the cloud, the GODNRRS technique is implemented in Eclipse IDE Java Language with Cloudsim simulator by using the two real application workflows namely LIGO and Montage. This technique employs a diverse number of user tasks in the range of 100-1000 from Cybershake, LIGO and Montage real-time application to accomplish experimental evaluation. When using 800 user tasks from cybershake application as input, this technique obtains 72J energy consumption whereas state-of-the-art works MPGA and JIT Workflow Scheduling and IRR acquires 152J, 104J, and 80 J respectively. From that, it is clear that the energy consumption using proposed technique is lower than other state-of-the-art works. The performance result analysis of energy consumption for LIGO and Montage, cybershake workflows is portrayed in below Table IV. Fig. 5 (a, b, c) describes the experimental results of energy consumption for Cybershake, LIGO and Montage real-time applications based on different number of user tasks using three methods namely MPGA, JIT Workflow Scheduling, IRR and GODNRRS technique.

As shown in Fig. 5, proposed technique obtains lower energy consumption to perform tasks in the considered real-time applications when compared to MPGA, JIT Workflow Scheduling and IRR.

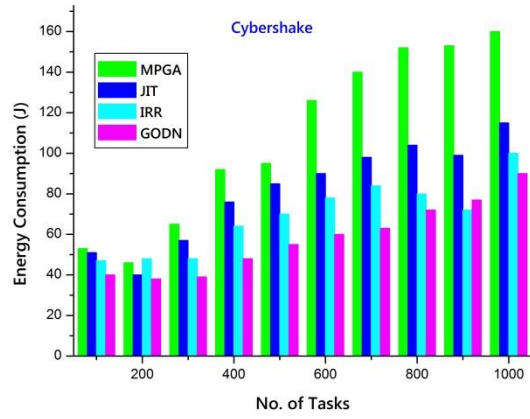
TABLE IV. TABULATION FOR ENERGY CONSUMPTION

Number of user tasks (N)	Energy consumption (J)											
	Cybershake Workflow				LIGO Workflow				Montage Workflow			
	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS	MPGA	JIT	IRR	GODNRRS
100	53	51	47	40	54	53	51	46	55	52	49	43
200	46	40	48	38	48	46	42	50	50	42	52	44
300	65	57	48	39	69	66	60	57	72	60	54	48
400	92	76	64	48	92	88	80	72	96	80	72	60
500	95	85	70	55	105	100	90	85	110	90	80	70
600	126	90	78	60	114	108	102	96	120	96	90	78
700	140	98	84	63	119	112	105	98	126	105	91	77
800	152	104	80	72	136	128	112	104	144	112	96	80
900	153	99	72	77	135	126	117	108	144	108	99	81
1000	160	115	100	90	140	130	120	115	150	121	118	112

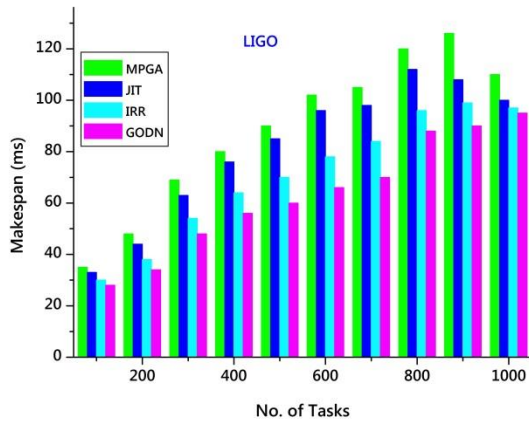




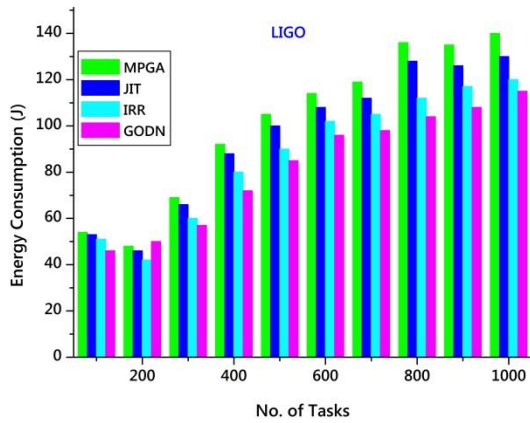
(a) Makespan Analysis for Cybershake Workflow



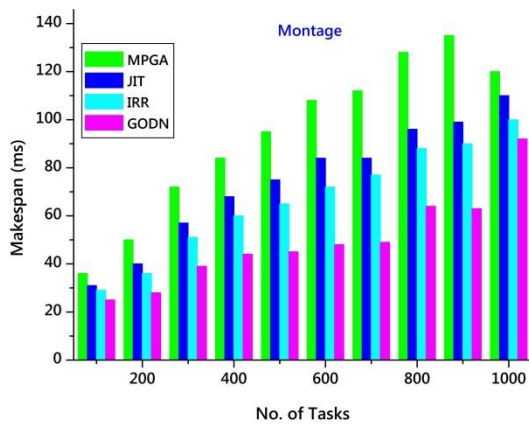
(a) Energy Analysis for Cybershake Workflow.



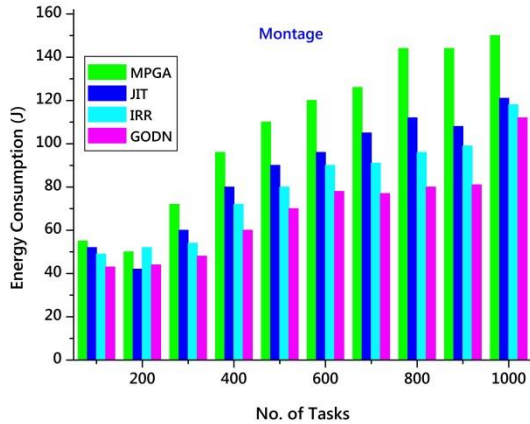
(b) Makespan Analysis for LIGO Workflow



(b) Energy Analysis for LIGO Workflow.



(c) Makespan Analysis for Montage Workflow.



(c) Energy Analysis for Montage Workflow.

This is because of the application of the genetic algorithm, round-robin scheduling and deep neural learning in proposed GODNRRS technique on the contrary to existing works. With the concepts of GODNRRS, proposed technique identifies optimal virtual machines based on fitness measurement (i.e. energy, memory, CPU time, bandwidth capacity) to carry out

user tasks in the cloud. This assists for proposed GODNRRS technique to minimize the amount of energy taken to effectively perform Cybershake, LIGO and Montage workflows in Cloud as compared to existing works. Therefore, the proposed GODNRRS technique minimizes the energy utilization by 16%, 12 % and 4 % as compared to MPGA, JIT

Workflow Scheduling and IRR respectively when considering LIGO real-time applications. Similarly, the proposed GODNRRS Technique decreases the energy consumption by 33 %, 19 % and 14% as compared to compared to MPGA, JIT Workflow Scheduling and IRR respectively when taking Montage real-time applications. Likewise, the proposed GODNRRS technique reduces the energy consumption by 43 %, 27 % and 16% as compared to compared to MPGA, JIT Workflow Scheduling and IRR respectively when taking cybershake real-time applications.

## VI. CONCLUSION AND FUTURE WORK

The GODNRRS technique is designed with the goal of increasing the scheduling performance of workflows in the cloud environment when employing a huge number of user tasks as input. The goal of GODNRRS technique is obtained with aid of a genetic algorithm, round-robin scheduling in deep neural learning on the contrary to existing works. The designed technique achieves enhanced scheduling efficiency while increasing the number of user tasks as input as compared to conventional works. Further, the proposed GODNRRS Technique attains lower makespan for scheduling all the workflows in the cloud environment as compared to state-of-the-art works. In addition to that, this technique obtains minimal energy, memory, CPU time, bandwidth utilization to perform the user jobs on cloud server as compared to existing works. The efficiency of proposed technique is estimated in terms of scheduling efficiency and makespan and energy utilization for both LIGO, Montage and cybershake real-time applications and compared with two existing works. The experimental result shows that GODNRRS technique presents a better performance with an improvement of scheduling efficiency and minimization of makespan when compared to state-of-the-art works. In future work, we are planning to deal with various workflows from different regions by considering different data centers. Finally our aim is to implement our proposed technique in a real time cloud environment.

## REFERENCES

- [1] Henrique Yoshikazu Shishido, Julio Cezar Estrella, Claudio Fabiano Motta Toledo, and Marcio Silva Arantes, "Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds", *Computers & Electrical Engineering*, Vol. 69, Pages 378-394, 2018.
- [2] J. Sahni and D. P. Vidyarthi, "A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment," in *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 2-18, 1, March 2018.
- [3] Indrajeet Gupta, Madhu Sudan Kumar, and Prasanta K. Jana, "Efficient Workflow Scheduling Algorithm for Cloud Computing System: A Dynamic Priority-Based Approach", *Arabian Journal for Science and Engineering*, Springer, Vol. 43, Issue 12, pp. 7945–7960, December 2018.
- [4] Maria Alejandra Rodriguez, and Rajkumar Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds", *IEEE Transactions on Cloud Computing*, Vol. 2, Issue 2, pp. 222 – 235, June 2014.
- [5] Aida A. Nasr, Nirmeen A. El-Bahnasawy, Gamal Attiya, and Ayman El-Sayed, "Cost-Effective Algorithm for Workflow Scheduling in Cloud Computing Under Deadline Constraint", *Arabian Journal for Science and Engineering*, Springer, vol. 44, Issue 4, pp. 3765–3780, April 2019.
- [6] R. Duan, R. Prodan and X. Li, "Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds," in *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 29-42, January 2014.
- [7] Farzaneh Abazari, Morteza Analoui, Hassan Takabi, Song Fu, "MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm", *Simulation Modelling Practice and Theory*, Vol. 93, pp. 119-132, May 2019.
- [8] Z. Zhu, G. Zhang, M. Li and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1344-1357, 1 May 2016.
- [9] Juan J. Durillo, Vlad Nae, and Radu Prodan, "Multi-objective energy-efficient workflow scheduling using list-based heuristics", *Future Generation Computer Systems*, Vol. 36, pp. 221-236, July 2014.
- [10] Uang-shun Yao, Yong-sheng Ding, and Kuang-rong Hao, "Multi-objective workflow scheduling in cloud system based on cooperative multi-swarm optimization algorithm", *Journal of Central South University*, Springer, Vol. 24, Issue 5, pp. 1050–1062, May 2017.
- [11] V. Arabnejad, K. Bubendorfer and B. Ng, "Budget and Deadline Aware e-Science Workflow Scheduling in Clouds," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 29-44, 1 Jan. 2019.
- [12] S. Esteves and L. Veiga, "WaaS: Workflow-as-a-Service for the Cloud with Scheduling of Continuous and Data-Intensive Workflows," in *The Computer Journal*, vol. 59, no. 3, pp. 371-383, March 2016.
- [13] Long Chen, and Xiaoping Li, "Cloud workflow scheduling with hybrid resource provisioning", *The Journal of Supercomputing*, Springer, Vol. 74, Issue 12, pp. 6529–6553, December 2018
- [14] Ehab Nabil Alkhanak, and Sai Peck Lee, "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing", *Future Generation Computer Systems*, Vol. 86, pp. 480-506, September 2018.
- [15] Zhiqiang Xie, Xia Shao, and Yu Xin, "A Scheduling Algorithm for Cloud Computing System Based on the Driver of Dynamic Essential Path", *PLOS ONE*, Vol. 11, Issue 8, pp. 1-19, August 2016.
- [16] Ahmad M. Manasrah and Hanan Ba Ali, "Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-17, January 2018.
- [17] Fengyu Guo, Long Yu, Shengwei Tian, and Jiong Yu, "A workflow task scheduling algorithm based on the resources fuzzy clustering in cloud computing environment", *International Journal Of Communication Systems*, Vol. 28, Issue 6, pp.1053-1067, April 2015.
- [18] Amandeep Verma, and Sakshi Kaushal, "Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud", *Journal of Grid Computing*, Springer, Vol. 13, Issue 4, pp. 495–506, December 2015.
- [19] Q. Wu, F. Ishikawa, Q. Zhu, Y. Xia and J. Wen, "Deadline-Constrained Cost Optimization Approaches for Workflow Scheduling in Clouds," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3401-3412, 1 December 2017.
- [20] Xiumin Zhou, Gongxuan Zhang, Jin Sun, Junlong Zhou, Tongquan Wei, and Shiyun Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT", *Future Generation Computer Systems*, Vol. 93, pp. 278-289, April 2019.
- [21] D. Chitra Devi and V. Rhymend Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks," *The Scientific World Journal*, vol. 2016, pp. 1-14, December 2016.