

# A Cuckoo-based Workflow Scheduling Algorithm to Reduce Cost and Increase Load Balance in the Cloud Environment

Shahin Ghasemi<sup>#</sup>, Ali Hanani<sup>\*</sup>

<sup>#</sup> Department of Computer Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

<sup>\*</sup> Department of Computer Engineering, Songhor and Koliaei Branch, Islamic Azad University, Songhor, Iran  
E-mail: shghasemi@kums.ac.ir, ali.hanani@iauksh.ac.ir

**Abstract**— Workflow scheduling is one of the important issues in implementing workflows in the cloud environment. Workflow scheduling means how to allocate workflow resources to tasks based on requirements and features of the tasks. The problem of workflow scheduling in cloud computing is a very important issue and is an NP problem. The relevant scheduling algorithms try to find optimal scheduling of tasks on the available processing resources in such a way some qualitative criteria when executing the entire workflow are satisfied. In this paper, we proposed a new scheduling algorithm for workflows in the cloud environment using Cuckoo Optimization Algorithm (COA). The aims of the proposed algorithm are reducing the processing and transmission costs as well as maintaining a desirable load balance among the processing resources. The proposed algorithm is implemented in MATLAB and its performance is compared with Cat Swarm Optimization (CSO). The results of the comparisons showed that the proposed algorithm is superior to CSO in discovering optimal solutions.

**Keywords**— Cloud Computing, Workflow, Scheduling, Optimization, Cuckoo Algorithm

## I. INTRODUCTION

Today, with the advancement of information technology (IT), it is necessary to do comprehensive computing (wherever and whenever). In addition, users should be able to do their heavy computing work without having expensive hardware and software. Cloud computing is the latest technology solution to meet these needs. The National Institute of Technology and Standards (NIST) defines cloud computing as follows [1]: *Cloud computing is a model for providing easy access to a variety of adjustable and configurable computing resources (such as networks, servers, storage spaces, applications, and services) based on user requests and through the network, so that access can be provided fast with minimal necessity of resource management or direct involvement of the service provider.*

Most organizations will be able to use various cloud services to facilitate their jobs by migrating workflows into the cloud computing environment. In general, a workflow models the running steps within a large job. So, many complex and heavy jobs can be expressed in the form of a workflow. This will simplify the scheduling of these jobs in addition to simplifying the process of running any task in the job. Therefore, workflow scheduling is one of the key issues

in managing workflow implementation, which attracted the attention of many researchers.

In short, workflow scheduling is the mapping of each task in a workflow to an appropriate processing source in such a way that dependencies among tasks are not violated and satisfies some qualitative criteria such as make-span, processing cost, transmission cost, load balancing, and quality of service [2, 3]. These requirements are usually expressed in the service level contract between the customer and the service provider. Also, it's clear that in different cloud computing environments, there are different service contracts. This raises the complexity of workflow scheduling process [4, 5]. Therefore, the problem of workflow scheduling in the cloud environment is an optimization problem that is investigated in this study.

Already, several meta-heuristics algorithms, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimization (GWO), Imperialist Competitive Algorithm (ICA), CSO, Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) have been used to solve various optimization problems. The COA is one of the most powerful meta-heuristics algorithms to solve optimization problems [6], which was first introduced in 2009.

In this study, a COA-based workflow scheduling algorithm is proposed to perform tasks in the cloud. The cuckoo search algorithm is a meta-heuristic optimization methodology that has an evolutionary approach in finding and exploration of optimized solutions. The cuckoo optimization algorithm is inspired by the amazing behavior of cuckoo breeding and combines it with a Levy flight method that is a random patrol. The Levy's flight method is a random walk that the length of its pace follows from a given distribution [7].

The rest of this paper is organized as follows. Section II reviews previous work and COA. Also, it presents the proposed scheduling algorithm. Section III discusses the performance evaluation and simulation results. The paper is concluded in Section IV.

## II. MATERIAL AND METHOD

In this section, we first review some existing workflow scheduling algorithms in the cloud. Then, we present the COA in details. Finally, the proposed scheduling algorithm is presented.

### A. Related Work

In [8], a PSO-based algorithm is proposed to schedule tasks to cloud resources. Particle Swarm Optimization is a swarm-based intelligence algorithm inspired by the social behavior of animals such as a bird's swarm looking for a food source or a bunch of fish that protect themselves from the hunter, etc. A particle in the PSO is like a bird or a fish in the search space. The motion of each particle is regulated by a velocity that includes amplitude and direction. The position of each particle at any time is influenced by its best position and the position of the best particle in a problem space. The performance of a particle is evaluated with fitness, which is specific to the problem space.

The PSO algorithm is like other evolutionary methods. The population in PSO is the number of particles in a problem space. Each particle has a fitness value calculated by a fitness function to be optimized for each generation. Each particle knows its best position ( $P_{best}$ ), and  $P_{best}$  is far from the best particle of the whole particle group ( $g_{best}$ ). The  $P_{best}$  of a particle is the best result (the amount of fitness) obtained by this particle, while  $g_{best}$  is the best particle in terms of the amount of fitness in the entire population. In each generation, the velocity and position of the particles are updated in each generation [8].

Bilgayan et al. [9] presented a CSO-based workflow scheduling in the cloud environment. The cat algorithm consists of two general steps: seeking step and tracing step. In the seeking step, the cats remain stationary and do not move. But in the tracing step, the cats move quickly to the next best position. In summary, a population of  $n$  cats is initially formed. Then, a random speed is determined for each cat. In the next step,  $p\%$  of cats in the population are randomly placed in the tracing step and the rest of the cats in the seeking step. Then, the fitness of each cat is calculated and the best cat's position is stored in the memory. In the next step, the position of the cats is updated according to their status (seeking or tracing). These steps are repeated to discover the optimal solution.

Also, Malosky et al. [10] presented a cost optimization model for the scheduling of scientific workflow in the cloud environment. In this model, it was assumed that several IaaS clouds with heterogeneous virtual machine samples with a limited number of samples per cloud and hourly billing are available. Input and output data are stored on a cloud such as the Amazon S3. The proposed model uses a computational programming language and allows users to minimize the cost of running workflow under time limits.

Navondo et al. [11] presented a new resource allocation and pricing model for batch work in the cloud system. Users in this model input tasks with a value function. Satisfaction with payment is a function of the debt history of the work. The cloud provider in response provides a subset of these tasks. Non-preemptive allocation of resources is among important points of this model.

Deepak et al. [12] also proposed a scheduling algorithm that, in order to reduce the implementation cost, schedules tasks in cloud resources using two different pricing models, while the time limitation of the workflow is observed. This algorithm is resistant to premature completion errors and cloud resource throughput fluctuations.

Alkhanak et al. [13] discussed the cost-aware workflow scheduling. The researchers initially provided an overview of concepts and research related to cost-aware scheduling, and then, classified cost-aware challenges based on service quality, system throughput, and system architecture for workflow scheduling in the cloud computing. The focus of this research is on the execution cost of the workflows, and the load balance is not considered here.

Zhang et al. [14] proposed a method for scheduling grid systems using the birds' movement algorithm. Although this article states that the scheduling goal is to increase resource throughput in addition to reducing make-span, the fitness function is considered only in the first criterion. Finally, the authors of this paper implemented their method using the genetic algorithm and after comparing the practical results of the algorithm of birds' movement with the genetic algorithm, it was concluded that the algorithm of the birds' movement is to produce better results.

Rimal et al. [15] used a cyclic & directed graph model to solve the problem of workflow scheduling in the cloud environment. Load balancing is the main purpose of this algorithm, so that more freely accessible workstations can be used. This algorithm relies on the best effort and tries to make the best possible decision.

Xave et al. [16] presented a multi-objective optimization method for workflow scheduling in a cloud environment. Make-span and productivity are the main criteria for optimization in this algorithm. A genetic method was used to design this algorithm. The throughput of this algorithm was compared with the algorithm of birds. The results show that the genetic algorithm has better throughput.

Pratiba et al. [17] also used the PSO algorithm to schedule workflows in medical software. A medical software can be implemented efficiently (in terms of cost) in the cloud service model. In this research, firstly, medical software is mapped as the service resources available in the cloud environment, and then, a discrete particle optimization method is used to minimize the cost of running the software. The scheduling generated by this algorithm will be re-

scheduled based on the running time, as well as the rating of financial resources.

Varma et al. [18] also focused on the multi-objective optimization problem of workflow scheduling in the cloud environment. The proposed scheduling algorithm in this study is based on a hybrid PSO method whose goal is to optimize two criteria: make-span and cost.

Goyal et al. [19] presented a hybrid algorithm based on two methods PSO and ACO. In this hybrid algorithm, parameters such as processor power, processor memory, the cost of running a task on a specific processor, the cost of communication links between processors and the cost of communication between the tasks are used.

Rodriguez et al. [20] presented a workflow scheduling algorithm for scientific workflows, based on short-term budgeting. In this algorithm, a detailed pricing model has been used that enables the user to avoid unnecessary use of the Internet. The main purpose of this algorithm is to reduce the make-span. To achieve this goal, criteria such as budget constraints and quality of service were used.

### B. Cuckoo Optimization Algorithm (COA)

The algorithm for the first time was introduced in 2009 by Shin-Oh Yang and Deb Savsh [6] and described in detail by Ramin Rajabyon in 2011 [7]. The cuckoo optimization algorithm is based on cuckoo's life, which tries to embed its eggs in the nest of other birds and impose egg hatching and breeding on other host birds. The flowchart of the cuckoo optimization algorithm is presented in Fig. 1.

This algorithm, like other evolutionary algorithms, starts with a primitive and random population of cuckoos. This initial population of cuckoos has eggs that embed them in the nest of host birds. Some of these eggs, which are more like host bird's eggs, will have a greater chance of growing and becoming mature cuckoos. The host birds may detect and eliminate cuckoo's eggs. The number of eggs that grow indicates the suitability of nests in that area. Cuckoos are looking for the best area to maximize their eggs' survival rates. After the chicks are born and converted to adult cuckoos, groups of cuckoos are formed. Each group lives in its own habitat. The best settlement area among all the groups will be the next destination for the cuckoos in other groups. All groups migrate to the best region. Then, based on the number of eggs per cuckoo and the distance of cuckoos from the current optimal area, a laying radius for each cuckoo is calculated. Then the cuckoos begin to lay randomly in the nests which are in their own laying radius. This process is repeated to make cuckoos discover the best habitat and laying area.

### C. The Proposed Algorithm

The main purpose of the proposed scheduling algorithm is to find optimal schedules for executing workflows in the cloud environment. The main idea of this proposed algorithm is to use the cuckoo algorithm to solve the problem of scheduling workflow. The optimization criteria in the proposed algorithm are:

- Cost minimization (including processing cost and transmission cost).
- Maximize the load balancing between processing stations

In summary, the following points are important in the process of workflow scheduling:

- A workflow consists of several smaller tasks that may or may not depend on one another. The workflow is usually represented as a directional graph with no cycle.
- Running any task requires several files or input data, and produces files as output.
- Approximate volume of input and output files is known for each job.
- A set of cloud virtual machines is available and the cost of executing any job on any virtual machine is known by calculation and statistical data.
- The cost of transferring data between virtual machines is known.

In the following, we describe the steps in the proposed algorithm in accordance with the flowchart of the cuckoo algorithm in Fig 1.

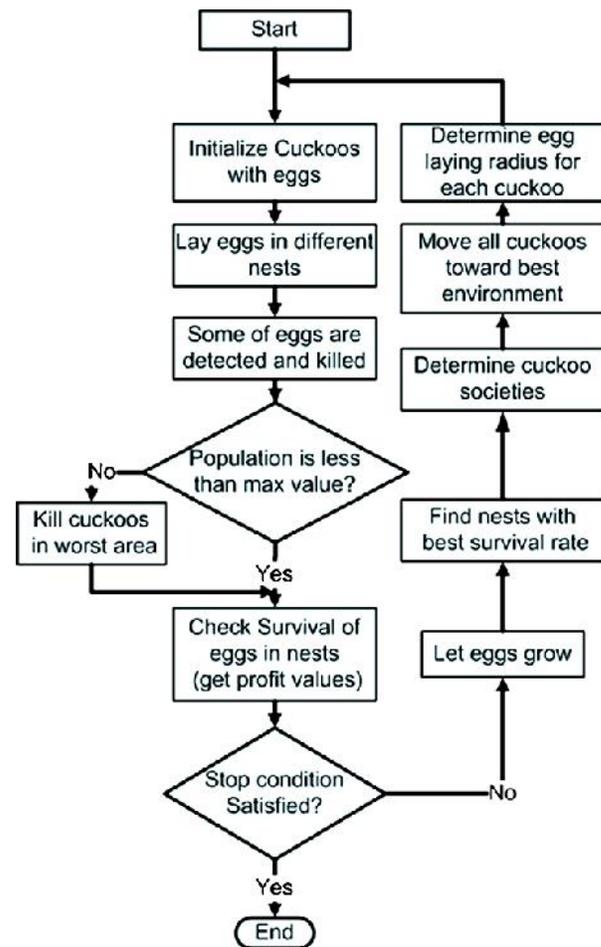


Fig. 1 Flowchart of the COA [7]

### C.1 Initializing Cuckoo Population

In the first step of the proposed algorithm, an initial population of cuckoos and their eggs are generated. The initial population of cuckoos is created randomly. In the proposed algorithm, each location (host's nest) is equivalent to a possible solution to the problem of scheduling. Each cuckoo or any cuckoo egg is in a habitat. Therefore, each cuckoo also represents a possible solution to the problem of scheduling.

Assuming that  $V = \{v_1, v_2, \dots, v_m\}$  is the set of processing stations or virtual machines assigned to executing the workflow  $T$  such that  $T = \{t_1, t_2, \dots, t_n\}$ , then each habitat  $h_i$  or cuckoo or cuckoo's egg is represented as an  $n$ -member vector, as equation (1). The first element of this vector determines task  $t_1$  is done by which VM. The second element states that task  $t_2$  is executed by which VM, and the last element states that task  $t_n$  must be executed by which VM.

$$h_i = [v_1, v_2, \dots, v_k] \quad , l, j, k \leq m \quad (1)$$

Then for each  $h_i$ , some eggs are produced. In nature, every cuckoo lay between 5 and 20 eggs. These numbers are used as the upper and lower limit of the allocation of eggs to each cuckoo in different repetitions. Another common habit of every real cuckoo is that it lays eggs in a certain range that is called the Egg Laying Radius (ELR). Fig. 3 shows how a cuckoo lay in the ELR.

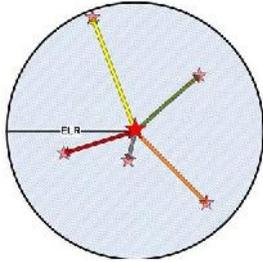


Fig. 2 An example of egg laying [7]

In an optimization problem, each variable has a high limit  $Var_{hi}$  and a lower limit  $Var_{low}$ , which each ELR can be defined using these limits. The ELR is proportional to the total number of eggs, the number of present eggs of the cuckoo, and the upper and lower limits of the problem variables. Therefore, ELR is calculated from equation (2) [7].

$$ELR = \alpha \times \frac{\text{Number of current cuckoo's eggs}}{\text{Total number of eggs}} \times (Var_{hi} - Var_{low}) \quad (2)$$

Here,  $\alpha$  is an integer that controls the maximum ELR value. Also, in the proposed algorithm, the  $Var_{hi}$  and  $Var_{low}$  of variables are proportional to the number of tasks in the workflow model. So  $Var_{low}=1$  and  $Var_{low}=n$ .

The egg-laying method in the proposed algorithm runs in this way: If we assume that there are four VMs, means  $V = \{v_1, v_2, v_3, v_4\}$ , then there are eight tasks in the workflow model, means  $T = \{t_1, t_2, \dots, t_8\}$ , and cuckoo  $h_i$  is located as follows:

$$h_i = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ \hat{v}_1 & \hat{v}_1 & \hat{v}_3 & \hat{v}_2 & \hat{v}_4 & \hat{v}_3 & \hat{v}_2 & \hat{v}_3 \end{bmatrix}$$

If the laying radius for this cuckoo is  $ELR=4$ , then a possible egg laying pattern for this cuckoo is as follows:

$$h_i = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ \hat{v}_2 & \hat{v}_4 & \hat{v}_3 & \hat{v}_2 & \hat{v}_4 & \hat{v}_3 & \hat{v}_3 & \hat{v}_3 \end{bmatrix}$$

In short, for each egg of cuckoo  $h_i$ , four ( $ELR=4$ ) elements of  $h_i$  is randomly selected and their mapping changes randomly.

### C.2 Eggs Being Killed by Host Birds

At this stage,  $p=10\%$  of all eggs produced by all cuckoos are randomly selected and removed. Each egg is a solution to the problem after becoming a chick and matured.

### C.3 Cuckoo's Death in Inappropriate Areas

In this step, if the number of cuckoos exceeds the maximum population ( $Max_{pop}$ ), the cuckoos are removed in inappropriate areas (low profit). Given that each cuckoo in the proposed algorithm represents a possible solution to the problem of scheduling, the cuckoos are removed from the population at inappropriate (or low profit) places.

### C.3 Fitness Evaluation

At this stage of the proposed algorithm, the fitness of each cuckoo is calculated. Different cloud service providers have provided several pricing policies to determine the cost of services in the cloud environment. For example, service provider Amazon has provided the Amazon Web Services AWS<sup>1</sup> calculator to calculate costs for its users. If each cuckoo's position is represented by a vector  $M$ , then, according to the equations (3) ~ (6) in the researches [8] and [9], the fitness of each cuckoo is obtained by equation (6). Equation (3) and equation (4) compute the processing cost and transmission cost imposed on each VM by scheduling  $M$ , respectively. Equation (5) calculates the total cost (the plural of the processing and transmission costs) imposed on each VM. Finally, the fitness of  $M$  is calculated by equation (6). In fact, the fitness of each cuckoo is equal to the maximum processing and transmission costs imposed on VMs.

$$C_{exe}(M)_j = \sum_k w_{kj} \quad \forall M(k) = j \quad (3)$$

$$C_{tx}(M)_j = \sum_{k1 \in T} \sum_{k2 \in T} d_{M(k1), M(k2)} \times e_{k1 k2} \quad (4)$$

$$\forall M(k1) = j \text{ und } M(k2) \neq j$$

$$C_{total}(M)_j = C_{exe}(M)_j + C_{tx}(M)_j \quad (5)$$

$$Cost(M) = \text{mux}(C_{total}(M)_j) \quad \forall j \in P \quad (6)$$

### C.4 Cuckoos' Evolution

When cuckoo chicks have grown and matured, they live for some time in their surroundings and groups, but when the laying time approaches, they will migrate to better habitats that have the chance of eggs survival. Following the formation of groups of cuckoos in different areas (problem search space), the group habitat with the best position is selected as the destination for the migration of other cuckoos. When adult cuckoos live in the environment, it's difficult to

determine which cuckoo belongs to which group. Various methods such as K-means can be used to group cuckoos. The number of groups is usually between 3 and 5 groups.

In groups created by the K-means algorithm, a group that is in a more appropriate area (i.e. it contains more suitable solutions) is selected as the optimal group for migration of the remaining cuckoos. In the proposed algorithm, the criterion for choosing the optimal group is that, first, the difference in the profit of each cuckoo (non-leader) is compared with the profit of the leader cuckoo. This difference in profit is calculated for all cuckoos in the group and summed up for each group and the average is determined. Each group whose average profit is lower than that of its leader is chosen as a better group.

Cuckoos do not go all the way to the target area when migrating to the optimal area. They only fly a part of the path and deviate in that too. This mode of movement is clearly seen in Fig. 3. Each cuckoo only flies % of the entire path to the target area and has radians diversion. These two parameters help the cuckoos to search for more environment. Random variable is between 1 and 0 and is a number between /6 and - /6 [7]. The formula for the migration operator in the cuckoo optimization algorithm is given by equation (7):

$$X_{NextHabitat} = X_{currentHabitat} + F(X_{GoalPoint} - X_{CurrentHabitat}) \quad (7)$$

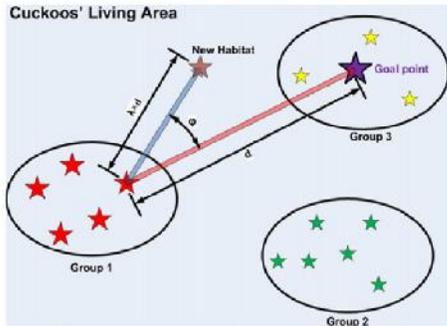


Fig. 3 cuckoos migrate method [7]

Here,  $F$  is a parameter that causes deviation.

### III. RESULTS AND DISCUSSION

The proposed algorithm was implemented by MATLAB and its throughput was compared with the CSO-based algorithm [9]. For experiments, the dataset and the workflow model presented in the [9] was used. There is a workflow of 17 tasks and five VMs in this dataset. The workflow of these tasks is shown in Fig. 4. The transmission cost between VMs is shown in Table 1 and Table 2 shows the processing cost of each task on VMs.

The maximum population size was considered  $Max_{pop}=50$ . The number of groups of cuckoos was also considered  $N_{Group}=5$ . In carrying out experiments, the throughput of the proposed algorithm and CSO was evaluated for 50 to 400 generations. Each experiment was repeated 50 times, and the final results were obtained from an average of these 50 times.

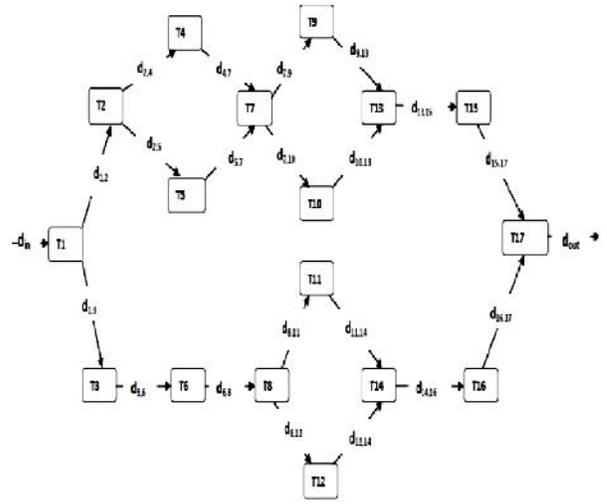


Fig. 4 the workflow model [9]

TABLE 1.  
THE COMMUNICATION COST BETWEEN VMs (CENTS PER MEGABYTE) [9]

	VM1	VM2	VM3	VM4	VM5
VM1	0	0.01	0.15	0.19	0.2
VM2	0.01	0	0.15	0.19	0.05
VM3	0.15	0.15	0	0.2	0.19
VM4	0.19	0.19	0.2	0	0.15
VM5	0.2	0.01	0.11	0.15	0

TABLE 2.  
THE PROCESSING COST OF ANY TASK ON EACH VM (CENT) [9]

	VM1	VM2	VM3	VM4	VM5
T1	1.23	1.12	1.15	1.13	1.16
T2	1.17	1.17	1.28	1.14	1.17
T3	1.13	1.11	1.11	1.13	1.16
T4	1.26	1.12	1.14	1.15	1.15
T5	1.19	1.14	1.22	1.15	1.15
T6	1.23	1.12	1.15	1.13	1.15
T7	1.13	1.11	1.11	1.13	1.11
T8	1.26	1.12	1.14	1.15	1.12
T9	1.11	1.11	1.13	1.15	1.13
T10	1.26	1.12	1.14	1.15	1.17
T11	1.17	1.17	1.28	1.14	1.17
T12	1.26	1.12	1.14	1.15	1.15
T13	1.23	1.12	1.15	1.13	1.14
T14	1.26	1.12	1.14	1.15	1.17
T15	1.26	1.12	1.14	1.15	1.15
T16	1.11	1.11	1.13	1.15	1.13
T17	1.17	1.17	1.28	1.14	1.17

In the first experiment, the effect of the size of transferring files (*DataSize*) between tasks was evaluated for the proposed and CSO algorithms and the results are presented in term of total cost. In the second experiment, the performance of the proposed algorithm is evaluated in term of load balance metric.

**Experiment 1:** In the first experiment, the proposed algorithm and CSO were evaluated for different size of transferring files between VMs, *DataSize*=64MB, 128MB, 256MB. The results of this experiment are shown in Fig. 5 for *DataSize*=64MB, Fig. 6 for *DataSize*=128MB, and Fig. 7 for *DataSize*=256MB, in terms of total cost.

The results of this experiment showed that the proposed algorithm was able to discover better solutions than the CSO algorithm. For example, in Fig. 5, after 50 generations, the best solution discovered by the proposed algorithm is the cost of 15.7, while the result for the CSO algorithm is about 18. This superiority continues to advance the proposed algorithm in generations of more than 50.

The reasons for the proposed algorithm's superiority are that the CSO algorithm tends to discover local optimal solutions, while the proposed algorithm focuses more on discovering global optimized solutions. Also, in the proposed algorithm, all cuckoos in inappropriate locations migrate toward the best area in each generation. So, the speed of convergence is greater.

Clearly, with the increase in the file size between VMs i.e. *DataSize*, the transmission cost increases. This will result in an increase in the total cost of running the workflow. The results of this experiment showed that increasing the size of transferring files in the workflow from 64MB to 128MB and 256MB the proposed algorithm was also able to find better solutions than the CSO algorithm. In this experiment, the proposed algorithm has discovered better solutions in less than 200 generations, and both algorithms have the same throughput when generation exceed 200.

**Experiment 2:** The second experiment aims to evaluate the performance of the proposed algorithm in terms of load balance. The results of this experiment are shown in Fig. 8. The results showed that the proposed algorithm distributes the jobs on VMs, in a desirable manner. The schedules that discovered by the proposed algorithm impose a balanced load on VMs.

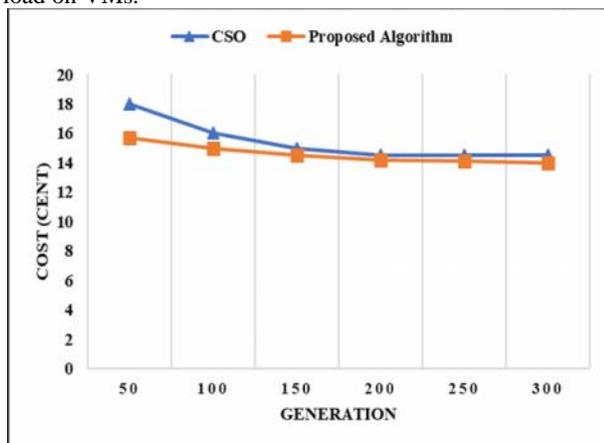


Fig. 5 Comparison of the proposed algorithm and CSO in terms of total cost for *DataSize*=64MB

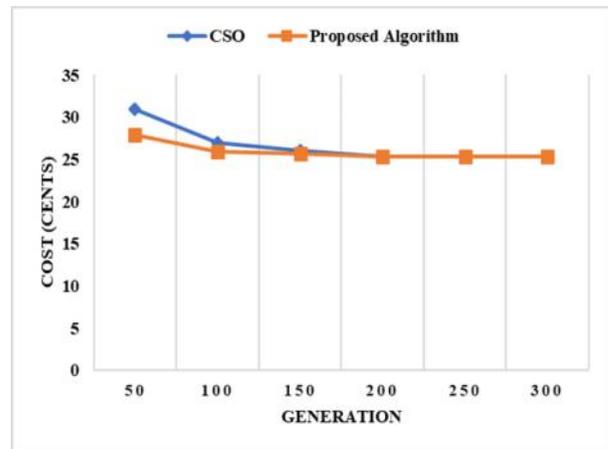


Fig. 6 Comparison of the proposed algorithm and CSO in terms of total cost for *DataSize*=128MB

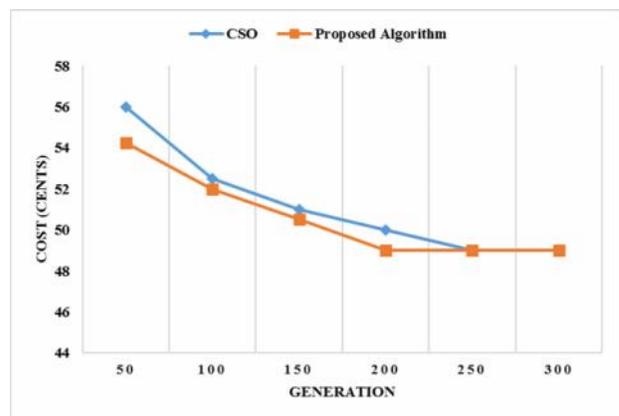
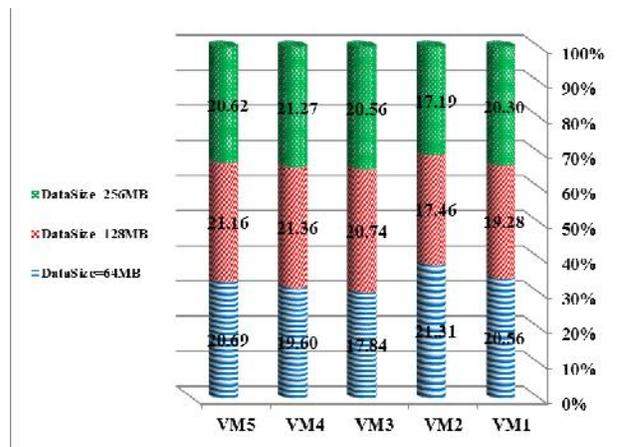


Fig. 7 Comparison of the proposed algorithm and CSO in terms of total cost for *DataSize*=256MB



#### IV. CONCLUSION

In this paper, a new algorithm for workflow scheduling in the cloud environments was proposed using a cuckoo meta-heuristic algorithm. The cuckoo algorithm (COA) is one of the new evolutionary algorithms that was first introduced in 2009. The main objective in designing the proposed algorithm is to optimize the cost of executing the entire

workflow and the load balance between the VMs. The proposed algorithm was implemented and its performance was evaluated in terms of cost and load balance criteria. The simulation results of the proposed algorithm were compared with the results of the CSO algorithm. The results of the comparisons show that the proposed algorithm succeeds in discovering the better solution, in lower generations.

#### REFERENCES

- [1] Mell, P. and Grance, T., 2009. The NIST definition of cloud computing, NIST special publication 800-145, National Institute of Standards and Technology. Available: <http://www.csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
- [2] Abolfazli, S., Sanaei, Z., Sanaei, M.H., Shojafar, M. and Gani, A., 2015. Mobile cloud computing: The-state-of-the-art, challenges, and future research.
- [3] Wang, J., Korambath, P., Altintas, I., Davis, J. and Crawl, D., 2014. Workflow as a service in the cloud: architecture and scheduling algorithms. *Procedia computer science*, 29, pp.546-556.
- [4] Ranjbari, M. and Torkestani, J.A., 2018. A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing*, 113, pp.55-62.
- [5] Bala, A. and Chana, I., 2011, November. A survey of various workflow scheduling algorithms in cloud environment. In *2nd National Conference on Information and Communication Technology (NCICT)* (pp. 26-30).
- [6] Yang, X.S. and Deb, S., 2009, December. Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210-214). IEEE.
- [7] Rajabioun, R., 2011. Cuckoo optimization algorithm. *Applied soft computing*, 11(8), pp.5508-5518.
- [8] Pandey, S., Wu, L., Guru, S.M. and Buyya, R., 2010, April. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on* (pp. 400-407). IEEE.
- [9] Bilgaiyan, S., Sagnika, S. and Das, M., 2014, February. Workflow scheduling in cloud computing environment using cat swarm optimization. In *Advance Computing Conference (IACC), 2014 IEEE International* (pp. 680-685). IEEE.
- [10] Malawski, M., Figiela, K., Bubak, M., Deelman, E. and Nabrzyski, J., 2015. Scheduling multilevel deadline-constrained scientific workflows on clouds based on cost optimization. *Scientific Programming* (pp. 5).
- [11] Jain, N., Menache, I., Naor, J.S. and Yaniv, J., 2014. A truthful mechanism for value-based scheduling in cloud computing. *Theory of Computing Systems*, 54(3), pp.388-406.
- [12] Poola, D., Ramamohanarao, K. and Buyya, R., 2014. Fault-tolerant workflow scheduling using spot instances on clouds. *Procedia Computer Science*, 29, pp.523-533.
- [13] Alkhanak, E.N., Lee, S.P. and Khan, S.U.R., 2015. Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems*, 50, pp.3-21.
- [14] Zhang, L., Li, K., Li, C. and Li, K., 2017. Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems. *Information Sciences*, 379, pp.241-256.
- [15] Rimal, B.P. and Maier, M., 2017. Workflow scheduling in multi-tenant cloud computing environments. *IEEE Transactions on Parallel and Distributed Systems*, 28(1), pp.290-304.
- [16] Zhu, Z., Zhang, G., Li, M. and Liu, X., 2016. Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on parallel and distributed Systems*, 27(5), pp.1344-1357.
- [17] Prathibha, S., Latha, B. and Suamthi, G., 2017. Particle swarm optimization based workflow scheduling for medical applications in cloud. *Biomedical Research*.
- [18] Verma, A. and Kaushal, S., 2017. A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. *Parallel Computing*, 62, pp.1-19.
- [19] Goyal, M. and Aggarwal, M., 2017. Optimize workflow scheduling using hybrid ant colony optimization (ACO) & particle swarm optimization (PSO) algorithm in cloud environment. *Int. J. Adv. Res. Ideas Innov. Technol.*, 3(2).
- [20] Rodriguez, M.A. and Buyya, R., 2017. Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 12(2), pp.5.