International Journal of Grid and Distributed Computing
Vol. 9, No. 7 (2016), pp.149-158
http://dx.doi.org/10.14257/ijgdc.2016.9.7.16

Scheduling Jobs on Cloud Computing using Firefly Algorithm

Demyana Izzat Esa , Adil Yousif

University of Science and Technology-Omdurman-Sudan
adiluofk@gmail.com

Abstract

Cloud computing is a new technology, instead of all computer hardware and software that used on desktop, or somewhere within company's network, it's presented as a service by cloud service providers and accessed via the Internet. Exactly where hardware and software are located and how everything works does not matter. In cloud computing there are many jobs that requires to be executed on the available resources to achieve best minimal execution time. Several optimization methods are available for cloud job scheduling. However, the job scheduling process is still need to be optimized. This paper proposes a new job scheduling mechanism using Firefly Algorithm to minimize the execution time of jobs. The proposed mechanism based on information of jobs and resources such as length of job speed of resource and identifiers. The scheduling function in the proposed job scheduling mechanism firstly creates a set of jobs and resources to generate the population by assigning the jobs to resources randomly and evaluates the population using a fitness value which represents the execution time of jobs. Secondly the function used iterations to regenerate populations based on firefly behavior to produce the best job schedule that gives the minimum execution time of jobs. Several scenarios are implemented using Java Language and CloudSim simulator. Different settings have been considered in the evaluation and experimentation phase to examine the proposed mechanism in different workloads. The first phase of the evaluation process describes how the proposed mechanism can be used to minimize the execution time of jobs. The second phase of the evaluation process compares the proposed mechanism with First Come First Serves (FCFS) algorithm. The results revealed that the proposed mechanism minimizes the execution time significantly. Furthermore, the proposed mechanism outperformed the FCFS algorithm.

Keywords: *Cloud, Job Scheduling, Metaheuristic, Firefly Algorithm*

1. Introduction

As the IT technologies are growing day by day, the need of computing and storage are rapidly increasing. Cloud computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer technology. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services [1].

Cloud Computing has become a widely accepted paradigm for high performance computing, because in Cloud Computing all type of IT facilities are provided to the users as a service. The services of the cloud are provided through the Internet. In Cloud Computing the term Cloud is used for the service provider, which holds all types of resources for storage, computing etc. Mainly three types of services are provided by the cloud. First is Infrastructure as a Service (IaaS), which provides cloud users the infrastructure for various purposes like the storage system and computation resources.

Second is Platform as a Service (PaaS), which provides the platform to the clients so that they can make their applications on this platform. Third is Software as a Service (SaaS), which provides the software to the users and hence the user don't need to install the software on their own machines and they can use the software directly from the cloud. Cloud Computing provides many benefits: it results in cost savings because there is no need of initial installation of much resource; it provides scalability and flexibility, the users can increase or decrease the number of services as per requirement; maintenance cost is very less because all the resources are managed by the Cloud providers[2].

Job scheduling is one of the major activities performed in all the computing environments. Cloud computing is one the upcoming latest technology which is developing drastically. To efficiently increase the working of cloud computing environments, job scheduling is one the jobs performed in order to gain maximum profit.

The goal of scheduling algorithms in distributed systems is spreading the load on processors and maximizing their utilization while minimizing the total job execution time Job scheduling, one of the most famous optimization problems [3]. Job scheduling has been considered as one of crucial problems in cloud computing. An optimized scheduler would improve many factors in scheduling of jobs in a cloud system such as throughput and performance. Different Approaches have tried to solve this problem like Genetic algorithm, Ant colony optimization, Particle swarm optimization and etc.

There are several resources on cloud and huge number of jobs, suppose that R= $\{r_1, r_2, r_3 \ldots r_m\}$ are m cloud resources and J= $\{j_1, j_2, j_3 \ldots j_n\}$ are n independent jobs. The speed of each resource is expressed in form of MIPS (Million Instructions Per Second), and the length of each job is expressed in the form of number of instructions. Now the problem is how to allocate the submitted jobs to the available resources in order to complete the jobs efficiently. The current scheduling mechanisms are still need to be optimized, such that an optimum execution time is achieved.

The objective of this paper is to propose a new job scheduling mechanism based on Firefly Algorithm with modified distance to minimize the execution time and to evaluate the proposed mechanism using CloudSim simulator.

This paper contains six sections. Section two describes the job scheduling on cloud computing. Section three describes the Firefly Algorithm. Section four illustrates the proposed scheduling mechanism. Section five is the evaluation and experimentation part. We concluded in section six.

2. Cloud Job Scheduling

One of the major concerns in cloud computing is job scheduling. It is an extensive research area in cloud computing. As in Figure (2), job scheduling of customers' tasks means how to allocate resources to these tasks. Thus, the required tasks can be accomplished in minimum time according to time defined in user request.

Most researches that used in grid computing can also be used in cloud computing setting. The central task of job scheduling system is to determine the most suitable resources for the user's jobs in a cloud computing. Scheduling in cloud computing can be divided into two main views, from the cloud computing users (CCU) and from the cloud computing service provider (CCSP).

From the user's point of view, the scheduling algorithm should generally reduce both the execution time and cost. On the other hand, the main concern of the cloud service provider is that the scheduling algorithm should improve the resource utilization and reduce the maintenance cost along with energy consumption[4].

3. Job Scheduling Methods

Cloud Computing system is a developing technology. Therefore, scheduling the tasks in the cloud is vitally important. A number of issues have to be considered which include improving the utilization of resource in cloud, minimizing the processing cost, increasing the performance of the server, and lastly minimizing the processing time and completion time. One of the main research fields is to develop a schedule mechanism of the tasks in cloud. This schedule mechanism was investigated by several researchers who have suggested various algorithms in order to provide better solutions[5].

3.1. Bees Life Algorithm

Salim Bitam presented a paper on new Bee Swarm optimization algorithm called Bees Life Algorithm (BLA) which is inspired by the nature life of bees. It is applied to efficiently scheduling computation jobs among processing resources onto the cloud data centers. Then, a set of experimental tests has been carried out with Genetic Algorithm (GA) to compare the two mechanism. The results demonstrate that BLA outperforms GA in terms of execution time (makespan) with the least complexity[6].

3.2. RSDC (Reliable Scheduling Distributed In Cloud Computing)

Arash Ghorbannia Delavar, et al proposed a consistent scheduling algorithm in cloud computing environment. In this algorithm the main job is divided into sub jobs. In order to balance the jobs, both the request and acknowledge times are calculated separately. The scheduling of each job is done by calculating the request and acknowledges times in the form of a shared job. Hence, the efficiency of the system is improved[7].

3.3 Deadline and Budget Distribution based Scheduling

The goal of this scheduling algorithm is to develop workflow schedule in a way that it reduces the execution cost and yet meet the time constraints demanded by the user, in order to overcome scheduling problems efficiently[8].

3.4 Improved Genetic Algorithm

Pardeep Kumar and Amandeep Verma in [2] presented a paper to boost genetic algorithm for allocation of resources to user requests. They proposed an idea for generating initial population by using Min-Min and Max-Min. These algorithms can suggest better initial population than randomly selecting the initial population. Consequently, the proposed idea contributes to an Improved Genetic Algorithm.

The author has used CloudSim as a simulator for checking the performance of both the newly improved Genetic algorithm and the standard Genetic Algorithm. The author used Virtual Machines as resource and Cloudlets as tasks/jobs. He tested the performance of these algorithms in two events: in the first event, he selected the fixed number of virtual machines and the varied number of cloudlets; in the second event he chose the fixed number of cloudlets and the varied number of virtual machines. The resource utilization was calculated in both cases.

The experiment resulted in better outcomes in the favor of improved genetic algorithm, where the utilization of resources was better than in the case of the standard genetic algorithm. Almost all the resources in the case of improved genetic algorithm were utilized in a better way, whereas with the standard genetic algorithm there was a large gap in the percentage of the resources usage.

This means that some resources in the standard genetic algorithm were fully utilized, while some others were not. On the other hand, using the improved genetic algorithm has resulted in minimizing the makespan, and at the same time utilizing the resources more efficiently, compared to the standard genetic algorithm.

In addition, the author used the makespan as fitness criteria for checking the fitness of the scheduling results. This idea can be further extended in which we can use execution cost of the resource as fitness criteria.

3.5 Ant Colony Optimization

The author in [9] proposed a policy on task scheduling based on the concept of Ant Colony Optimization. The original ACO was designed to resolve the traveling salesman problem, by altering some aspects of the ACO to be accustomed to task scheduling problem and using simulator CloudSim. This was achieved by extending the Datacenter Broker class for designing the proposed scheduling policy based on ACO, and using makespan (the total finish time of the tasks) to evaluate the performance of the default policy and the proposed policy. The result demonstrated that the scheduling policy based on ACO can minimize the makespan of the tasks submitted to the cloud system.

3.6 ACO-LB Algorithm

In 2014 Shengjun Xue et al, in[10]presented a paper on ACO-LB (Load balancing optimization algorithm based on ant colony algorithm) algorithm to solve the load imbalance of virtual machine in the process of task scheduling. It was proposed that the ACO-LB algorithm will not only shorten the makespan of task scheduling, but also maintain the load balance of virtual machines in the data center. They compared their proposed ACO-LB algorithm to a number of other algorithms, which included the First-Come-First-Served (FCFS) scheduling strategy, the basic ant colony algorithm (ACO), and a combination algorithm (ACO-RE) of ACO and roulette wheel algorithm. This was carried out to test whether the ACO-LB algorithm retains more superior performance and load balancing ability than others. This paper used CloudSim to simulate a cloud data center and override the Datacenter Broker class and the Cloudlet class to analyze the simulation of these algorithms.The results confirmed that the proposed ACO-LB algorithm possess better performance and load balancing ability.

3.7 Improved Particle Swarm Optimization (PSO) algorithm

Shaobin Zhan and Hongying Huo, in[11]presented a paper to the improve particle swarm optimization algorithm in resources scheduling strategy of the cloud computing by merging between PSO algorithm and simulated annealing algorithm. Through experiments, the results indicated that this method can reduce the task average running time and increase the availability rate of resources. the authors added that, in future, they will lay particular stress on resource load balance of dynamic task scheduling in cloud computing, and consider other resource in cloud computing.

4. Firefly Algorithm

Fireflies are small winged nocturnal beetles that, during courtship, produce an intermittent light from luminescent chemicals in their abdominal organs.The firefly algorithm (FA) is a metaheuristic algorithm and its inspiration is based on light emission , light absorption and mutual attractiveness behavior among fireflies.

Initially, it was developed to solve optimization problems, but later on it was used for solving discrete problem and was also used in the fields of digital image processing, compression and clustering.

The FA was first presented by Xin-She Yang in late 2007 at Cambridge University. The algorithm derives its inspiration from flashing behavior of fireflies. The three idealized rules in describing FA algorithm[12, 13].

- All fireflies are attracted by each other without considering their sex.
- The attractiveness of fireflies is directly proportional to the brightness of them, it reduces with the increase in the distances between the fireflies. Normally a less bright one will move towards the brighter one. They will move randomly in case of the firefly having the highest bright.
- If there are no fireflies brighter than a given firefly, it will move randomly.

5. Proposed Firefly Algorithm for Cloud Job Scheduling

In the proposed mechanism the study used Firefly Algorithm in solving the problem of job scheduling and allocation of jobs to resources. Each firefly is a solution for allocation of jobs $\overrightarrow{X_{ij}}$, $i = (1,2,3,\ldots,n)$ $j = (1,2,3,\ldots,k)$, each element inside the firefly population vector is a random number between 1 to s where:

s is the total number of resources.

n is number of fireflies.

k is number of jobs that represent the length of each firefly.

The study represents resources as a vector that stores the speed of each resource $\overrightarrow{R_i}$, $i = (1,2,3,\ldots,s)$ and also jobs as a vector that stores the length of each job $\overrightarrow{J_i}$, $i = (1,2,3,\ldots,k)$, then we calculated the fitness function $F(\overrightarrow{X_{ij}})$ for each firefly by dividing each job length by the resource speed that the job is allocated to. The next step is to find the summation of the division results. This followed by finding the maximum fitness value. The firefly that has maximum fitness either moves randomly or does not move at all. The distance between each two fireflies is the number of non-corresponding elements[14] in the firefly population is calculated and stored in $\overrightarrow{D_{ij}}$, $i = (1,2,3,\ldots,n)$ $j = (1,2,3,\ldots,n)$ vector, and then calculate the attractiveness $\overrightarrow{\beta_{ij}}$, $i = (1,2,3,\ldots,n)$ $j = (1,2,3,\ldots,n)$ for each firefly from the fitness of the firefly by the equation $\overrightarrow{\beta_{ij}} = F(\overrightarrow{X_{ij}}) * e^{-\gamma \overrightarrow{D_{ij}}^{2}}$,where γ is fixed light absorption coefficient and e is exponential constant.

Finally firefly moves towards the brightest based on the attractiveness by the equation $x_{ij} = x_{ij} + \beta_{ij} e^{-\gamma D_{ij}^2} (x_i - x_j) + \propto (\text{rand} - \frac{1}{2})$

Where $\propto$ is randomization parameter between 0 and 1.

5.1 Pseudo Code for the Proposed Firefly Algorithm

Begin

```
Initialize parameter: t, iter_max, $\propto$, $\gamma$.
Generate initial population of fireflies $\overrightarrow{X_{ij}}$, $i = (1,2,3,\ldots,n)$ $j = (1,2,3,\ldots,k)$.
Set maximum iteration number=iter_max.
Set t=1
For each resource do
        Set speed for each resource $\overrightarrow{R_i}$
end for
for each job do
```

```
        Set length for each jobs $\vec{J_i}$
end for
while (t<=iter_max)
        for each firefly i do
                Compute Fitness function $F(\overrightarrow{X_{ij}})$
        end for
        foreach firefly ido
                for each firefly jdo
                        Compute the distance between firefly i and firefly j $\overrightarrow{D_{ij}}$
                end for
        end for
        for each firefly i do
                for each firefly j do
                        Compute the attractiveness of firefly i in the eye of firefly j
                        $\overrightarrow{\beta_{ij}} = F(\overrightarrow{X_{ij}}) * e^{-\gamma \overrightarrow{D_{ij}}^2}$
                end for
        end for
        for each firefly i do
                for each firefly j do
                        find the max attractiveness and its position
                end for
                for each job to firefly i do
                        Move firefly i towards firefly has max attractiveness using
                        $x_{ij} = x_{ij} + \beta_{ij} e^{-\gamma D_{ij}^2} (x_i - x_j) + \propto (rand - \frac{1}{2})$
                end for
        end for
        t←t+1
end while
```

6. Evaluation and Experimentation

To evaluate the proposed Firefly mechanism for cloud job scheduling this study implemented the algorithm using CloudSim simulator. Different scenarios are experimented to study various aspect of the mechanism. Execution time is used as measurement as illustrated previously. The experimentation phase scenarios are simulated as presented in the related works. The experiments setup and configuration and the values of the algorithm parameters are assigned based on the literature of the algorithm.

The simulation scenario used in this chapter considers the following four cases, in the first case, the number of jobs is 70, the number of resources is 30 and the number of fireflies is 15. In the second case, the number of jobs is 150, the number of resources is 80 and the number of fireflies is 50. The third case, a comparison of the execution times between Firefly Algorithm and First Come First Serves (FCFS) with the same number of jobs and resources (90 jobs and 50 resources) is conducted. The fourth case, compares the five execution times between Firefly Algorithm and FCFS with the different number of jobs and resources. This study selected FCFS mechanism to compare with as it considered the practical scheduling algorithm that is used widely in cloud data centers because of its simplicity.

5.1 The First Scenario

In this scenario the study considered a number of 70 jobs and 30 resources.

Table 1. The Execution Time of Ten Iterations in First Scenario

Iteration	1	2	3	4	5
Execution Time	254.8741868	254.8741868	254.8741868	204.9270992	204.9270992
Iteration	6	7	8	9	10
Execution Time	204.9270992	186.5182355	186.5182355	186.5182355	186.5182355

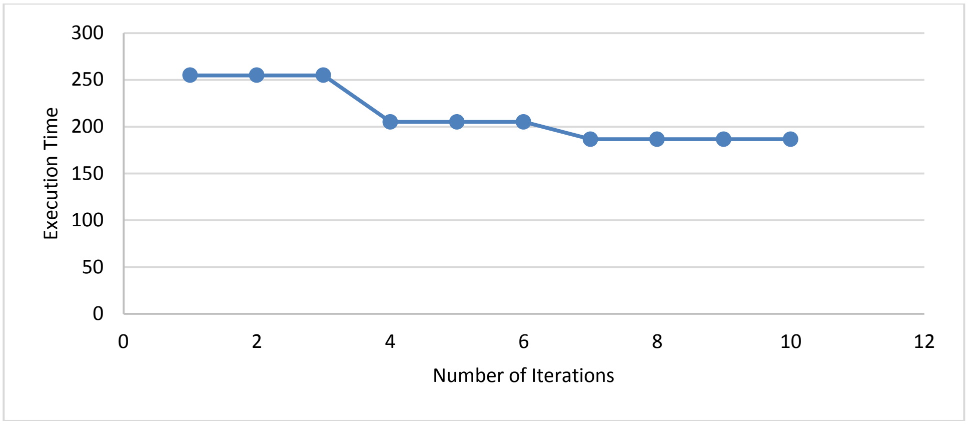

Figure 1. CloudSim Simulation Result in First Scenario

As described in Table (1) and Figure (1), the result of the initial execution time was 254.8741868 gradually decreased until it reached 186.5182355. This indicates that the proposed Firefly Algorithm minimized the execution time.

5.2 The Second Scenario

In this scenario the study considered a number of 150 jobs and 80 resources.

Table (2): The Execution Time of Ten Iterations in Second Scenario

Iteration	1	2	3	4	5
Execution Time	496.3979	496.3979	491.416	491.416	491.416
Iteration	6	7	8	9	10
Execution Time	491.416	491.416	458.3808	458.3808	458.3808

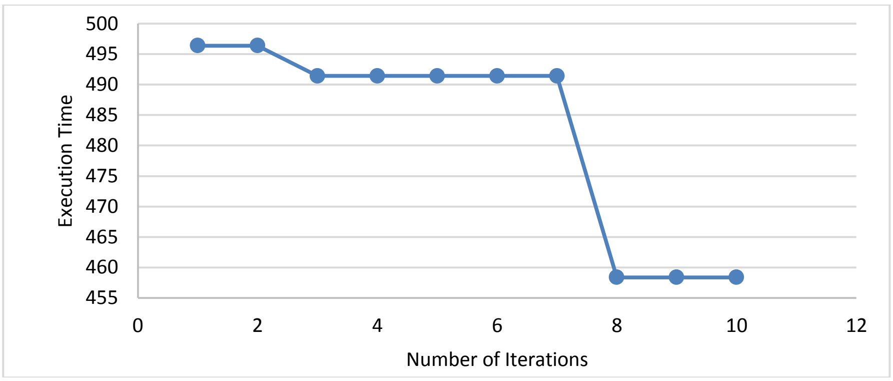

Figure 2. CloudSim Simulation Result in Second Scenario

As described in Table (2) and Figure (2), the result of the initial execution time was 496.3979 gradually decreased until it reached 458.3808.

5.3 The Third Scenario

This scenario compares the execution time between the proposed Firefly Algorithm and FCFS with the same number of jobs and resources (90 jobs and 50 resources).

Table (3): The Comparison between Firefly Algorithm and FCFS

Algorithm	Execution Time
FCFS	303.9271519
Random solution	408.0714958
Firefly Algorithm	205.3654806

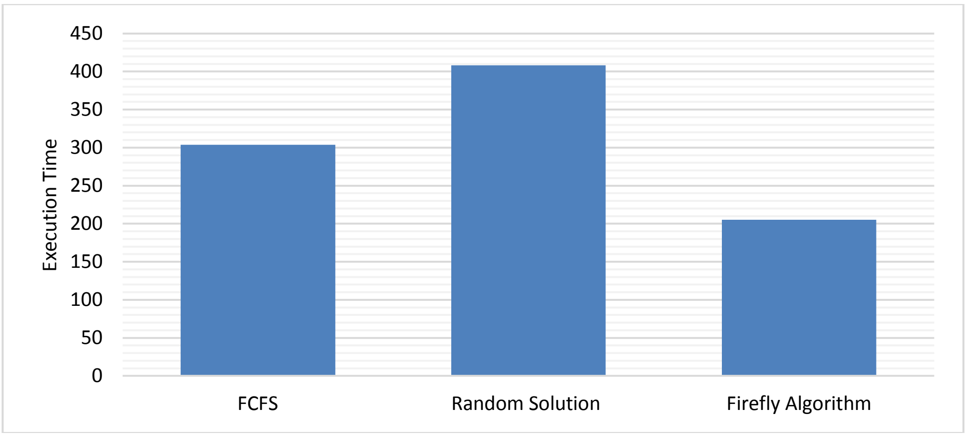

Figure 3. Comparison of Simulation Result in Third Scenario

As shown in Table (3) and Figure (3), Firefly Algorithm performed better than FCFS Algorithm based on the execution time. And the random solution has the worst results.

5.4 The Fourth Scenario

In this scenario, different numbers of jobs and resources are considered to evaluate the proposed mechanism and to examine the performance of the proposed mechanism in different workload and from different perspectives.

Table 4. Compare Five Execution Time between Firefly Algorithm and FCFS

Time	FCFS Algorithm	Firefly Algorithm
Time 1 (20 jobs 10 resources)	49.28968253968254	36.77023809523809
Time 2(60 jobs 30 resources)	179.84961392761173	159.95150399102292
Time 3(100 jobs 50 resources)	324.96026691647126	327.87379464872436
Time 4(120 jobs 80 resources)	462.28344311491	323.26934173035613
Time 5(150 jobs 100 resources)	599.9205338329427	337.2661870735075
Average Time	323.260708066323646	237.0262131077698

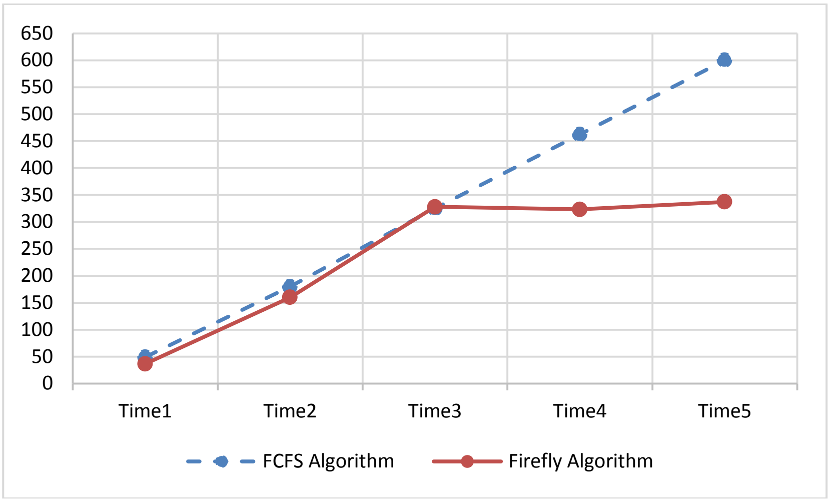

Figure 4. Comparison of Simulation Result in Fourth Scenario

As shown in Table (4) and Figure (4), the average of the execution time is calculated for five executions. The Firefly Algorithm outperformed FCFS Algorithm based on the execution time.

The previous results reveal that the proposed Firefly Algorithm for job scheduling in cloud optimizes the fitness value. As in all scenarios the fitness increase significantly which indicate that the execution time is decreased and optimized.

As shown in Figure (4), the proposed Firefly Algorithm has shorter execution time than FCFS when the number of jobs is small. When the number of jobs increased the difference in the execution time between FCFS and the proposed mechanism is increased significantly. This indicates that the proposed mechanism is more efficient when there is a heavy work load.

6.1. Conclusion and Future Work

This research proposed a new job scheduling mechanism to solve the scheduling problems by minimizing the execution time of jobs using Firefly Algorithm (FA).

The research work started with a mapping process to fitting the cloud job scheduling problem with the Firefly Algorithm. In this mapping process each job scheduling solution represents a firefly. A random number of solutions have been selected to represent the initial firefly population. The execution time of each schedule is considered the fitness function of the schedule. In each iteration, the schedules move based on the attractiveness of the schedule to generate a new population with enhanced fitness. To find the best solutions in each iteration, this research chooses the schedule that has the best fitness. The proposed Firefly Algorithm has minimized the execution time for the submitted jobs. Obviously, when increasing the number of jobs the execution time increases. However, the execution time of proposed mechanism is still less than First Come First Serves (FCFS).

References

[1] R. Kaur and S. Kinger, "Analysis of Job Scheduling Algorithms in Cloud Computing," *International Journal of Computer Trends and Technology (IJCTT),* vol. 9, pp. 379-386, 2014.

[2] P. Kumar and A. Verma, "Scheduling using improved genetic algorithm in cloud computing for independent tasks," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, 2012, pp. 137-142.

[3] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *IJRET: International Journal of Research in Engineering and Technology, ISSN,* pp. 2319-1163, 2013.

[4] M. Maqableh, H. Karajeh, and R. e. Masa'deh, "Job Scheduling for Cloud Computing Using Neural Networks," *Communications and Network*, vol. 2014, 2014.

[5] L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task scheduling optimization in cloud computing based on heuristic algorithm," *Journal of Networks*, vol. 7, pp. 547-553, 2012.

[6] S. Bitam, "Bees Life Algorithm for job scheduling in cloud computing," in *International Conference on Computing and Information Technology. ICCIT*, 2012, pp. 186-191.

[7] A. G. Delavar, M. Javanmard, M. B. Shabestari, and M. K. Talebi, "RSDC (reliable scheduling distributed in cloud computing)," *International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol*, vol. 2, 2012.

[8] A. Verma and S. Kaushal, "Deadline and budget distribution based cost-time optimization workflow scheduling algorithm for cloud," in *of the IJCA on International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT'12)*, 2012, pp. 1-4.

[9] L. Wang and L. Ai, "Task Scheduling Policy Based on Ant Colony Optimization in Cloud Computing Environment," in *LISS 2012*, ed: Springer, 2013, pp. 953-957.

[10] S. Xue, M. Li, X. Xu, and J. Chen, "An ACO-LB Algorithm for Task Scheduling in the Cloud Environment," *Journal of Software*, vol. 9, pp. 466-473, 2014.

[11] S. Zhan and H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing," *Journal of Information & Computational Science*, vol. 9, pp. 3821-3829, 2012.

[12] A. Yousif, S. M. Nor, A. H. Abdullah, and M. B. Bashir, "A Discrete Firefly Algorithm for Scheduling Jobs on Computational Grid," in *Cuckoo Search and Firefly Algorithm*, ed: Springer, 2014, pp. 271-290.

[13] X.-S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, pp. 36-50, 2013.

[14] A. Kumbharana and G. M. Pandey, "Solving travelling salesman problem using firefly algorithm," *International Journal for Research in Science & Advanced Technologies*, vol. 2, pp. 53-57, 2013.