A Hybrid Task Scheduling Approach using Firefly Algorithm and Gravitational Search Algorithm

Kanika Sood
M. Tech. Student, Computer Science Department,
Chandigarh Engineering college, Landran Mohali
kanika04sood@gmail.com

Dr. Anurag Jain
Assistant Professor (SG), Virtualization Department
School of Computer Science and Engineering
University of Petroleum & Energy Studies, Dehradun
anurag.jain@ddn.upes.ac.in

Dr. Amit Verma
Professor & Head, Computer Science Department
Chandigarh Engineering College, Landran, Mohali
hodcse@cecmohali.org

***Abstract*—With the meaningful improvements in Information Technology, it is envisioned that computing facilities will be considered as a 5th utility of the life after the four main utilities i.e. Water, gas, electricity and telephony. This computing utility, will contribute the fundamental stages of computing jobs that is necessary to fulfill all the demands of the public. Various computing prototypes have been introduced and cloud plays an important role. In this environment research has been done in the different fields like task scheduling, security, etc. The main question in this context is connected to task scheduling. This paper gives a hybrid task scheduling approach named as Hybridized firefly gravitational search algorithm (HFGSA). HFGSA approach is tested using Cloudsim simulator. The results of HFGSA approach has been compared with FA, GSA and Ant colony optimization algorithm (ACO). Improvement in processing time, response time and makespan is the major outcome of the HFGSA approach.**

***Keywords*—Cloud computing; load balancing; Cloudsim; HFGSA**

I. INTRODUCTION

Computing is being modified to a model that includes services delivered in a way equivalent to conventional utility such as water, electricity, gas and telephone. In October 2007, Google and IBM published collaboration with computing and it became more popular from then on. With the Amazon EC2, the web, email, Google App Engine and Sales force's CRM show an auspicious conceptual foundation of cloud services. It is a paradigm that provides convenient, on-request access to the network to serve computer recourses such as networks, servers, storage, applications and services that can be quickly supplied and release with least management work and service provider interaction [4,11]. Resources can be accessed via the web and a pay-per-use concept is offered via vendors. Anyone can easily register to Cloud services, and deploy and configure servers for creating an application in hours, and paying only for the time these resources have been accessed [4]. Cloud computing provides the gathered common and infrastructure services. It is a paradigm that provides all IT services to the users via internet. In this the word cloud is a collection of all types of resources such as servers, applications, storage, etc. Cloud offers primarily three categories of services. First service given by the cloud is an infrastructure as a service (IaaS), which offers infrastructure to the users. Second category of service is Platform as a service (PaaS), which offers a platform for the users to make their application. The third type of service is Software as a service that offers software to the users to run their applications directly from the cloud. Cloud computing has many advantages that are making it very useful day by day: it is cost effective because installation of most of the resources is not required; less maintenance charges because services are controlled by cloud vendors. Flexibility and scalability, resources are provisioned and released, according to the user's requirements.

In cloud, load balancing is a technique that distributes dynamic load across multiple systems to ensure no single system is overloaded [21]. In this paper task scheduling techniques are discussed. Existing task scheduling techniques have been analyzed in section II under the heading literature review. Motivation for proposed Hybridized firefly gravitational search algorithm (HFGSA) has been discussed in section III. Details of simulator used to test and compare the proposed HFGSA has been given in section IIV. Detail discussion of HFGSA has been given in section V. It is followed by section VI which covers the result and analysis. At the end, conclusion of paper has been given in section VII.

II. LITERATURE REVIEW

Shri Vratt Jawed Ahmed et al. in [9] proposed resource allocation algorithms and task scheduling that provides better results in a cloud computing environment. In order to allocate and manage the cloud system effectively, task scheduling is used. The resource allocation scheme can be used as a mechanism in a cloud computing system which aims to ensure that the application requirements are properly and correctly fulfilled by the provider's infrastructure.

978-1-5386-1887-5/17/$31.00 ©2017 IEEE

The Author in [10] offered an agreement on scheduling that depends on the idea of the Ant country in the supervision of another making the most out of. The first form ACO was designed to get separated the journeying person called travelling salesman problem whose work is the exchange of goods for money hard questions, by making a change some aspects of the ACO to be gotten used to job scheduling hard question and using simulator Cloudsim. This was achieved by getting stretched out the datacenter middleman part of designing the offered listing details agreement based on ACO, and using make span to value the activity of the Default agreement and the offered agreement. The outcome puts examples on view of that, the job scheduling agreement based on ACO can improve the overall performance in the cloud system.

Jiang, Ji et al. in [15] introduced a hybrid PSO and GSA (HPSO–GSA) in which position updates of particles are based on PSO velocity and GSA acceleration. Economic emission load dispatch problems were the main issue. HPSO-GSA gives better performance compared to PSO and GSA.

Han et al. in [16] introduced a QBGSA–K-NN algorithm in which they combined the quantum inspired Binary GSA (QBGSA) with K-nearest neighbor (K-NN) method with leave-one-out cross validation (LOOCV). It is used for improving classification accuracy with an appropriate feature subset in binary problems. It selects the discriminating input features correctly and achieves a high classification accuracy.

Sombra et al. in [3] introduced a new GSA in which changes were applied in alpha parameter throughout the iterations. It helps in improving performance of GSA. It helps to achieve better convergence.

Gu & Pan in [6] proposed a modified a GSA (GSA &PSO) in which PSO features of saving previous local optimum and global optimum solutions are implemented into GSA. The particle memory ability in GSA is modified to remember its own local optimum and global optimum solutions in the updating process. It gives better performance than SVM and SGSA-SVM in classification accuracy and feature selection ability.

A. *Firefly algorithm*

Fireflies are the small insects that giving off light in the dark with hard wing produce a light from luminescent chemicals in their stomach-related organs [5]. FA takes an inspiration from the natural process of these fireflies like light emission, common attraction behavior among insects and light absorption. Xin-She Yang in 2007 proposed this algorithm. It focuses on the following things.

- Fireflies are unisex it means they will attract others in spite of the gender.
- More is the brightness, more will be the attractiveness. Firefly with the lesser brightness will communicate with the firefly has more light. The brightest among all will walk unconsciously.
- To get the optimum result solution the illumination of a firefly is controlled by the objective function.

1) Firefly algorithm evaluation: FA solves the problem of distribution of jobs and task scheduling. In this algorithm for the distribution of the job every firefly is a solution.

$$C_{ij},\ i = (1,\ldots\ldots,f),\ j = (1,\ldots..,l)$$

Where elements in this vector are numbers between 1 to r where: r is the total resources, f represents the fireflies and l is the jobs that are the dimension of every firefly.
Resources and jobs have taken in the form of vectors, i.e. A_i , where i= (1,2,….,r) and B_i , where i= (1,2,….l). A vector contains the speed of the resources and B vector contains the dimension of the jobs. After this find the fitness function
F (C_{ij}) of every firefly is evaluated. This is evaluated by dividing the dimension of the jobs by the speed of the resources of the task. A distance of two fireflies is calculated and preserved in D_{ij}, i= (1,….,f), j = (1,…,f). Next step is to calculate the attractiveness of the fireflies from their fitness function F (C_{ij}).
Equation for evaluating the attractiveness:

$$\beta_{ij} = F\,(C_{ij})^{*}\, e^{-\gamma D2\, ij},$$

Where e is exponential constant and γ is light absorption constant. Firefly will walk among the brightest firefly by equation:

$$cij = cij + \beta ije\text{-}\gamma D2ij\ (ci - cj) + \alpha\ (rand\text{-}1/2)$$

Here each firefly is the combination of a cloudlet and virtual machine. Intensity of best firefly will be the function of selected parameter of virtual machine. Here the best intensity value means which cloudlet is best suited to the virtual machine.

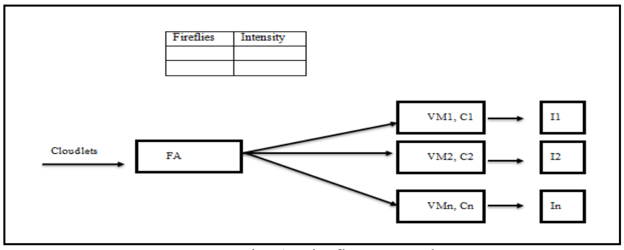

Fig. 1. Firefly approach

Firefly Algorithm is given below.

```
Algorithm FA ( )
{
  Configuration of virtual machines and cloudlets
   {
     Configure cloudlet for every virtual machine before submitting it
      While (calculating the cost of cloudlet w.r.t every virtual machine)
          If (cost is evaluated)
          {
            Evaluate intensity of each firefly
          }
            Choose best firefly among all
            {
              If (Intensity is best among all)
```

```
            {
                Make the best firefly of that intensity
                Execute the firefly
                Remove submitted cloudlet from waiting list
            }
          }
      }
}
```

B. *Gravitational search algorithm*

GSA was introduced by Rashedi in 2009. This algorithm obeys the rule of gravity and interaction of mass. The algorithm includes a collection of searcher agents that communicate with others via the weight force . The agents are denoted as things and their operation is measured by their masses. The gravity force causes a complete united attempt to get something done where all things move towards other things with greater weight masses. The slow attempt of greater weight masses gives good results [14].

The Gravitational search algorithm is given below:

```
Algorithm GSA ( )
{
  Configuration of virtual machines and cloudlets
   {
     Configure cloudlet for every virtual machine from the queue
      While (calculating the cost of cloudlet w.r.t every virtual machine)
        If (cost is evaluated)
        {
          Assign cloudlet to the virtual machine
        }
          For (position of the virtual machines)
          {
           Find best cloudlet-virtual machine pair
           Execute the pair
          }
            if (Position of old pair < current      pair)
              {
              The pair is in the first position
               Remove submitted cloudlet from     waiting list
              }
   }
}
```

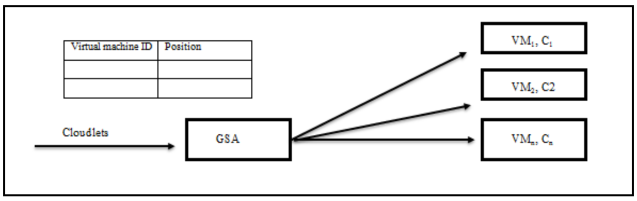

Fig. 2. Working of GSA

III. MOTIVATION

After reading all the existing approaches, the author had found the basic idea of a hybrid approach in [2]. In [2], authors have proposed a hybrid approach by incorporating the best features of join idle queue[1], join shortest queue[1] and minimum completion time[1]. So, the idea given in the existing techniques[2] has inspired authors to implement a new hybrid approach by including the best features of the selected existing approaches. The authors have selected two algorithms firefly and gravitation approach to make a hybrid approach. The Results of these algorithms are tested and compared using the Cloudsim simulator.

IV. CLOUDSIM SIMULATOR

When any approach is created, it is very important to check the feasibility of that approach. To check the feasibility of an approach, it is simulated in the simulator. In this paper Cloudsim has been used to simulate the proposed approach. Buyya et al. [17, 7] proposed a simulator called as Cloudsim which simulates all the applications and infrastructure that are available on a cloud. Cloud computing applications, infrastructures, allocation policies, service brokers etc. are ssimulated by the Cloudsim. To start a simulation, Cloudsim project has some important steps. It is necessary to know all these steps to work in Cloudsim [13]. There are steps that are followed in each example with some variation in them, specified as follows:

- Select the users that are proportional to number of brokers.
- Initialization of common variables
- Create CIS i.e. Cloudsim information service
- Cloudlet(s) are created with characteristics like Number of cloudlets (jobs/task), length, file size, output size, No of PE's required for execution of job/task.
- The Data center is created. It consists of characteristics like storage size, RAM size, MIPS, Bandwidth, No. of processing elements etc.
- Broker instance is created that is responsible for the communication between the datacenter and Vm and cloudlets.
- Virtual Machine(s) are created that includes characteristics like storage size, RAM size, MIPS, Bandwidth, No. of processing elements etc.
- Virtual Machine is submitted to the broker.
- Cloudlet(s) are created with characteristics like Number of cloudlets (jobs/task), length, file size, output size, No of PE’s required for execution of job/task.
- Cloudlets are submitted to the broker.
- Start Simulation.
- Stop Simulation, when execution is completed.

V. HFGSA APPROACH

In the HFGSA the study used the two algorithms that are named as Firefly algorithm and Gravitational search algorithm. The proposed algorithm is a hybrid approach and is implemented by using these two algorithms. The purpose of this proposed algorithm is to reduce the makespan and to maximize the overall performance. This proposed algorithm is simulated in Cloudsim simulator. The best intensity value has obtained from the Firefly Algorithm and is used as an input in the Gravitational search algorithm. To check the performance of our proposed algorithm, we have considered the concepts of ACO, Firefly algorithm, Gravitational search algorithm and hybridization of firefly and gravitational search algorithm.

Step1:The first step in the implementation of HFGSA algorithm is to configure the virtual machines and cloudlets. The window in figure 4 is used to set the configuration of virtual machine over the data center. Basic configuration of machine need: storage size, RAM size, MIPS (million instruction per second), Bandwidth, No. of processing elements, and name of the machine. Input parameters for generating cloudlet which also require some basic configuration which includes Number of cloudlets (jobs/task), length, file size, output size, No of PE's required for execution of job/task.

Step2: After configuring the cloudlets and virtual machine, if virtual machine or cloudlet is not created, the execution of all the cloudlets will be postponed. When virtual machines are created, configure the cloudlet for every virtual machine before submitting it.

Step3: In this step apply FA to evaluate best firefly. In HFGSA each firefly is the pair of a cloudlet and a virtual machine. E.g. F1= VM1_C1, F2= VM2_C1, F3=VM1_C2 and so on. After making pairs, the next step is to find the best pair, i.e. best firefly among all.

By applying FA, best firefly is evaluated by calculating the intensity of each firefly. To evaluate the intensity of each firefly, the cost of each cloudlet with respect to every virtual machine is calculated. A pair with the least cost among all is said to have best intensity and according to FA best intensity value is evaluated by the formulae

$$r = (F1-F0)^2$$

$$F1= F0 + \beta ije^{-\gamma D2ij} + (Rand-1/2)$$

Where Fo is Current required CPU speed, γ is CPU time to run cloudlet and Beta i.e. $\beta=e^{\gamma * r}$

In HFGSA intensity is the function of the selected parameters of the virtual machine and cloudlet. Best intensity of the firefly among all is evaluated.

Step4: Apply GSA, take the best intensity values evaluated from FA and use it as an input in the G value of the GSA. The Intensity has been used to get the best cloudlet-virtual machine pair to be executed. In GSA best pair is evaluated by getting the position. The Position is calculated by applying the law of gravity and law of motion. In HFGSA mass 1 is the product of processing element of virtual machine and MIPS required for the processing elements of that virtual machine and mass 2 is the product of the available MIPS of the virtual machine and current requested MIPS of that virtual machine.

m1 = Processing elements of a virtual machine* MIPS for processing elements

m2 = Available MIPS of VM* current requested MIPS

r= Square of the difference between the pairs of virtual machine and cloudlets

Mass = (mass1+mass2)/2;

Force = G*mass1*mass2/(r*r);

Acceleration =force/mass;

Velocity = acceleration;

Position = velocity*r;

Step5: Best pair is evaluated by getting the position of each cloudlet-virtual machine pair. After this best pair is executed and submitted cloudlet is removed from the queue.

```
Algorithm HFGSA ( )
{
  Configuration of virtual machines and cloudlets
  While (configure a virtual machine and cloudlet)
  {
    If (virtual machine and cloudlets are not created or virtual machine is not available)
    {
      Postpone the execution of all the cloudlets
    }
    Else
    {
      Configure cloudlet for every virtual machine before submitting it
    }
  }
      While (calculating the cost of cloudlet w.r.t every virtual machine)
      {
            If (cost is evaluated)
            {
              Calculate first firefly i.e. F0
              Calculate F1= F0 + βije^(-γD2ij) + (Rand-1/2)
              Calculate β=e^(γ*r)
              Calculate r= (F1-F0)^2
              Evaluate intensity of each firefly
            }
      }
        Compare the intensity of one firefly with another
        {
          If (Intensity is best among all)
            {
              Make it as an input for G value to the GSA
              Find the position of each firefly
              Calculate Mass = (mass1+mass2)/2;
              Calculate Force = G*mass1*mass2/(r*r);
              Calculate Acceleration =force/mass;
              Calculate Velocity = acceleration;
              Calculate Position = velocity*r;
              Find the best cloudlet-virtual machine pair
              Execute the pair
              Remove submitted cloudlet from waiting list
            }
        }
}
```

Figure 3 represents all the steps of HFGSA that is given below:

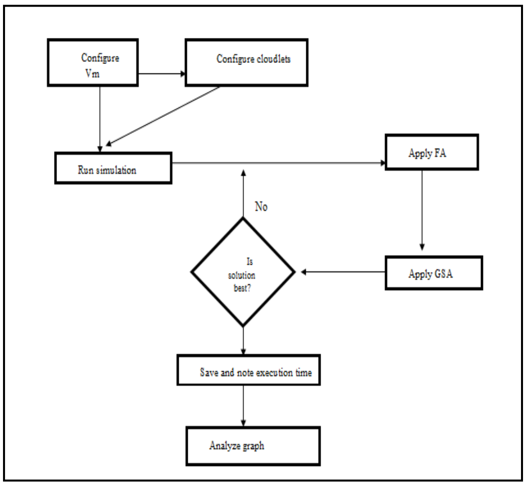

Fig. 3. HFGSA approach

VI. RESULTS

The results of HFGSA approach are shown in the form of the graphs. The proposed approach has been compared with other approaches.

A. Simulation configuration of VM1 and CC1

In these scenarios VM stands for virtual machine and CC is cloudlet configuration. Basic configuration of the virtual machine 1: Storage size=500 MB, RAM size=256 MB, MIPS (million instruction per second) =250, Bandwidth=1000, No. of processing elements=4, Name of the machine=Xen.

Basic configurations of cloudlet 1 that are used in HFGSA: Number of cloudlets (jobs/task) =100, Length=15000, File size=15000, Output size=1000, No of PE's required for execution of job/task=3.

B. Simulation configuration of VM2 and CC2

The basic configuration of the virtual machine 2:

Storage size=1000 MB, RAM size=512 MB, MIPS (million instruction per second) =200, Bandwidth=1500, No. of processing elements=3, and name of the machine=Xen.

The basic configurations of cloudlet 2: Number of cloudlets (jobs/task) =200, Length=20000, File size=2000, Output size=15000,

No of PE's required for execution of job/task=2

C. Results

The comparison between the execution time of these algorithms is given below. The performance of these algorithms has been measured in different environments. The Author has already discussed the configurations of virtual machine and cloudlets. So to get better results author has tested various algorithms in different environments. The results of this proposed algorithm are presented in the form of graphs. All the results are implemented in simulation kit named as Cloudsim. Figure 4 shows the comparison of various algorithms with the proposed algorithm by taking 9 virtual machines. Table 1 shows the performance comparison of algorithms with proposed algorithm.

Table I. Comparison of algorithms with proposed algorithm

No of tasks	ACO	FA	GSA	HFGSA
50	2266.55	2599.94	2066.47	466.65
150	5132.84	5199.51	8732.49	1333.29
250	8532.59	6732.67	15465.17	2333.15
350	11532.9	11065.57	22064.47	3266.35

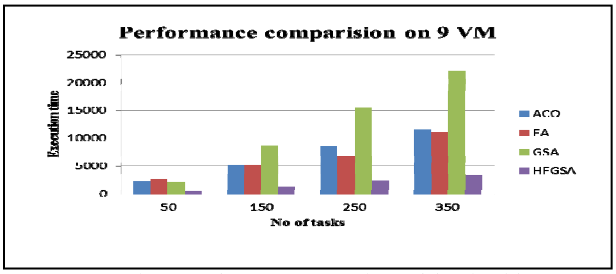

Fig. 4. Performance comparison of the algorithms

The comparison of a proposed algorithm has been done with the existing algorithms. This graph shows that the execution time of HFGSA is lesser than other algorithms. In First and second scenario the configurations of virtual machines are fixed and cloudlet configurations are varied.

1) The first scenario: In VM1_CC1, execution time of the algorithms is given as: ACO=3975. 02, FA=3300.13, GSA=4860. 1 and HFGSA=1200. 09.

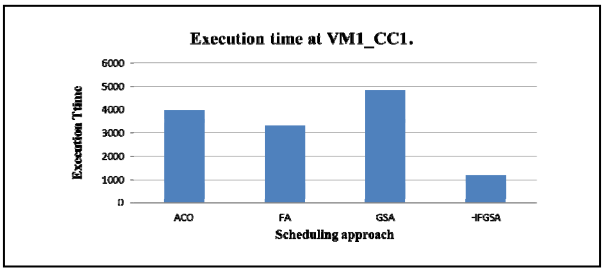

Fig. 5. VM1_CC1

*2) The second scenario:*FA=6599.94, GSA=16177.48, ACO=6933.27, HFGSA=2266.71

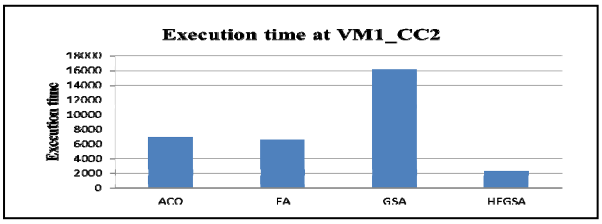

Fig. 6. VM1_CC2

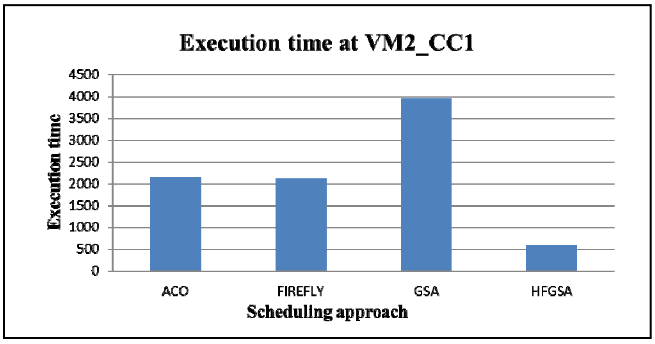

Fig. 7. VM2_CC1

3) The third scenario: Now in third and forth scenarios cloudlet configuration is fixed and virtual machine configurations are varied. ACO=3975.02, FA=3300.13, GSA=4860.1 and HFGSA=1200.09. These results are shown in figure 7.

4) The fourth scenario: Execution time of the algorithms: FA=4750.09,GSA=7240.02,ACO=5400.07,FGSA=1400.17

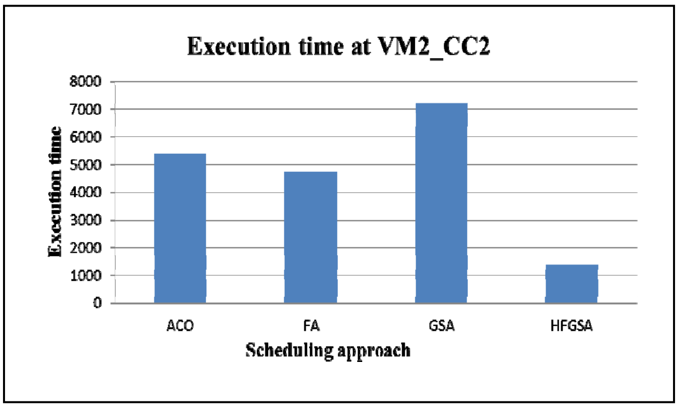

FIG. 8. VM2_CC2

VII. CONCLUSION

In this paper, a new job scheduling mechanism is proposed to solve the scheduling problems by minimizing the execution time of jobs. In this, hybridization of firefly and gravitational search algorithms for achieving cloud computing task scheduling has been presented. Firstly, the best intensity values of Firefly algorithm are evaluated by using simulation tool Cloudsim. Then, this intensity value is used as an input in the gravitational search algorithm. Simulation results demonstrate that the performance of hybridization of firefly and the gravitational search algorithm is better than other algorithms. A Comparison between these algorithms in different environment shows that the execution time of the proposed mechanism is less than another algorithm. From the present work, it has been observed that the results obtained using proposed algorithm is highly satisfactory compared with other techniques.

References

[1] A. Jain, R. Kumar "Critical Analysis of Load Balancing Strategies for Cloud Environment", Special Issue on: Advances in Mobile Cloud Computing and Social Networking of International Journal of Communication Networks and Distributed Systems. (SCOPUS Indexed), pp. 213-234, April 2017.

[2] A. Jain, R. Kumar, "Hybrid Load Balancing Approach for Cloud Environment", Special Issue on: Advances in Mobile Cloud Computing and Social Networking of International Journal of Communication Networks and Distributed Systems, (SCOPUS Indexed), pp. 264-286, April 2017

[3] A. Sombra, F. Valdez, P. Melin, and O. Castillo, "A New Gravitational Search Algorithm using Fuzzy Logic to Parameter Adaptation," IEEE Congress on Evolutionary Computation, pp. 1068– 1074, June 2013.

[4] B. Sosinsky, Cloud computing bible, John Wiley & Sons, Vol. 762, December 2010.

[5] D.I.E, Azir,"Scheduling Jobs on Cloud Computing Using Firefly Algorithm," (Doctoral dissertation, UNIVERSITY OF SCIENCE AND TECHNOLOGY), 2015.

[6] Gu and F. Pan, "Modified Gravitational Search Algorithm with Particle Memory Ability and its Application," International Journal of Innovative Computing, Information and Control, vol. 9, pp. 4531–4544 2013.

[7] K. Das, "Cloud computing simulation," PhD diss., Indian Institute of Technology, Bombay Mumbai, 2011.

[8] K. Dasgupta, B. Mandal, P. Dutta, J.K. Mandal, and S. Dam, "A genetic algorithm based load balancing strategy for cloud computing," Procedia Technology, pp. 340-347, Jan 2013.

[9] L.SV. Singh, J. Ahmed, "A Greedy Algorithm For Task Scheduling & Resource Allocation Problems In Cloud Computing," International Journal of Research & Development in Technology and Management Science–Kailash 21, 2014.

[10] 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.

[11] P.Mell, and T. Grance, "The NIST definition of cloud computing," 2011.

[12] R. Buyya, R. Ranjan and R.N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," In Proceedings of the 7th High Performance Computing and Simulation Conference, pp.1-11, June 2009.

[13] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience, Jan 2011.

[14] Sharma, Aakanksha. "DIFFERENTIAL EVOLUTION-GSA BASED OPTIMAL TASK SCHEDULING IN CLOUD COMPUTING." International Journal of Engineering Sciences & Research Technology 1, no. 5: 1447-1451.

[15] S. Jiang, Z. Ji, and Y. Shen, "A novel hybrid Particle Swarm Optimization and Gravitational Search Algorithm for Solving Economic Emission Load Dispatch Problems with Various Practical Constraints," InternationalJournal of Electrical Power & Energy Systems, vol. 55, pp. 628– 644, Feb 2014.

[16] X. Han, L. Quan, X. Xiong, and B. Wu, "Facing the Classification of Binary Problems with a Hybrid System Based on Quantum-Inspired Binary Gravitational Search Algorithm and K-NN Method," Engineering Applications of Artificial Intelligence, vol. 26, pp. 2424–2430, Nov 2013.