Received: June 20, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimized Scheduling and Resource Allocation Using Evolutionary Algorithms in Cloud Environment

Anusha Bamini Antony Muthu[1*] **Sharmini Enoch**[1]

[1]*Noorul Islam University, India*
* Corresponding author's Email: anushabamini@gmail.com

Abstract: Cloud computing is a powerful computing technology, which render a flexible services at anywhere to the user. One of the major issue of cloud computing was scheduling. In this work, a bacterial foraging optimization algorithm with genetic algorithm (GABFO) was combined to find out trustworthy scheduling problems in cloud workflow. Generally job scheduling and resource allocation in cloud is a tedious optimization problem at the time of considering QoS requirements. Lot of existing works under scheduling only concentrates on cost optimization and deadline problems, and it ignores the importance of reliability, availability and robustness. The main subscription of my work is to state a new optimized approach to schedule the jobs efficiently and allocate the resources in a efficient manner by introducing GABFO algorithm. Experiments were done in PSO, Genetic, BFO and then Genetic and BFO was combined to generate a hybrid optimized result, and the work was compared with above mentioned algorithms. The algorithms were executed for 52 iterations and totally 10 runs are calculated. The size of the job as well as virtual machines was varied for each iteration to calculate performance variation. We considered the optimization parameter as time and cost, and throughput. The work is implemented in cloudsim to create a simulated cloud environment. Final result shows better performance and maximum resource utilization in GABFO when compared to PSO, GA, BFO.

Keywords: Workflow, Scheduling, Resource allocation, Bacterial foraging optimization.

1. Introduction

In recent years distributed environments are playing a major role for computing. In that way cloud computing was used as a technology to use the resources based on pay and use model. Cloud services are classified in the terms of: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1, 2]. In this work we target on Infrastructure as a Service to utilize the pool of virtual resources. These resources are accessed on the basis of on demand. Resources considered in cloud are RAM, network speed, bandwidth. We can use the resources in a flexible manner. But the maximum utilization of resources provides better performance. The new scheduling techniques must be utilizing the resource efficiently.

To overcome lot of the problem in scheduling [3, 4, 5, 6, 7] a new workflow model was introduced in this paper. However the simple scheduling and resource allocation is not important, but the optimization of resource allocation [8, 9, 10] is important. Hence we developed a new scheduling approach which schedule the given job into number of tasks and allocate each task to resources in an optimized manner. In our work the first stage is to split the job into number of tasks, and in the second stage specific allocation of tasks into resources was done. The resource allocated is considered as virtual machines. Each job is allocated virtually to needed resources. For better allocation we combined the concept of genetic (GA) [11, 12] and bacterial foraging optimization (BFO). The fitness value is calculated for each iteration and the best fitted value was choose by considering minimized time and cost. The problem is to assign a task to resources and regulate the tasks on resources to optimize the entire cost and utilization time. The main objective is task scheduling and resource allocation must be carefully analyzed and jointly optimized to achieve reduced time and cost, and finally producing the QoS parameter of better reliability.

This work is based on the swarm intelligence techniques of bacterial foraging optimization (BFO) [13] and evolutionary computing concept of genetic algorithm (GA) [14-16]. BFO was introduced by the person Passino and it is inspired by the social foraging behavior of Escherichia coli. BFO has inspired the concentration of researchers for the reason that of its suitability and performance in recovering real-world optimization problems occurred in numerous application domains. The concept of E.coli is used to solve simple optimization problems. Where the chemotactic step size is adjusted for each run according to the current fitness of a bacterium. GA was inspired by Darwin's theory about Evolution. GA was commonly used for natural selection and generating high quality solutions in optimization. The objective of GA is to solve optimization problem.

In this paper, we develop a cost and time minimization scheduling technique which supports in the cloud. From the services of cloud computing we are using the feature of Infrastructure as a Service (Iaas). In IaaS we consider the features of a computing resources and virtual machine performance. To get an optimized result in task scheduling and resource allocation both concepts are merged. BFO and GA were combined to solve problems in scheduling and resource mapping. By using these algorithms the optimized results were achieved with good resource utilization.

The rest of the paper structured as follows. Section 2 provides a brief literature survey about scheduling, resource allocation concepts and various algorithms. Section 3 represents the problem definition, which includes the input, output, constraint and objective. Section 4 and 5 represents the standard bacterial foraging optimization algorithm and standard genetic algorithm. The proposed GABFO algorithm was explained in section 6. Simulation and analytical results were analyzed and plotted in section 7. Finally section 8 represents the conclusion of the entire work.

2. Related work

Scheduling under distributed systems has been studied well in the previous decades. Various algorithms are implemented in scheduling to meet the QoS constraints of users. The general aim of scheduling is to reduce the execution time of jobs. More number of scheduling algorithms is proposed for distributed computing [17-20]. Most of the algorithms [20-22] are applied for cloud scheduling based on its suitability. Goal of the scheduling algorithms are achieving better performance. The concept of scheduling the bag-of-task (BoT) application was proposed [23] in agent based scheduling concept. In this paper 14 scheduling concepts are executed concurrently. Based on the size of the task time is allocated for sharing the resources. The proposed elastic resource allocation technique will dynamically allocate and reallocate the resources. The result shows that the BoT was allocated and reallocated efficiently. The precedence constrained scheduling of parallel applications on heterogeneous computing systems (HCSs) was proposed in [5]. This proposes a parallel bi – objective hybrid genetic algorithm to reduce the energy consumption and increase makespan. The energy consumption was minimized by using a method of dynamic voltage scaling (DVS). Results prove that it dominates previous algorithm in terms of completion time, makespan and energy consumption.

A scheduling algorithm based on berger model [24, 25] was designed to establish the dual fairness constraint in virtualized cloud. The fairness of resource allocation was judged by the application of justice function. The results showed that the user tasks and the fairness were efficiently executed. A Biogeography-Based Optimization (BBO) was proposed to sole binary integer problem in job scheduling through better solution adaptation strategy [26]. In BBO, the GA and ACO strategies were incorporated to generate a new set of solutions, at each iteration; the Mann-Whitney test was conducted to evaluate performance output of BBO algorithm. Results proved that BBO performance was better than the GA and PSO algorithms. An Improved Genetic Algorithm (IGA) was proposed for job scheduling by speeding up the process of GA [27]. The proposed model has five components such as preprocessing unit, job schedulers, users, and data center and data center manager. The preprocessing unit encoded the attributes into users' job attribute vector, which included expected instruction count, job deadline and delay cost.

An optimal task scheduling and resource allocation was proposed using Particle Swarm Optimization (PSO) based fitness function [28-29, 31]. To balance the load PSO based fitness function was applied to reduce the make span and to maximize the processing capacity. The results showed that the PSO based method resulted in less execution time and cost. A Position Balance Parallel Particle Swarm Optimization (PBPPSW) [31] method was introduced with high profit resource allocation and flexible user satisfaction level was maintained. The performance metrics of average response time, total profit and number of virtual machines were considered for evolution. The results showed that the PB-PPSO method achieved increased profit and small response time with a less number of virtual machines.

A hybrid scheduling algorithm [32-34] was proposed by combining genetic algorithm and fuzzy theory to assign the task to virtual machines. Genetic algorithm was modified to balance the load and to reduce the execution time and cost. An agent based best fit resource allocation scheme was proposed to increase the resource utilization [35]. The results showed that the best fit approach was better in terms of job execution time, cost, virtual machine allocation and resource utilization. A continuous resource allocation strategy was presented to optimize the scheduling process in cloud [36]. The suggested resource allocation mechanism adopted minimal domination matching to compensate the trade-off space.

3. Problem definition

Assume the cloud customer has different jobs and each job is splitter into number task and each task has divided into sub tasks. These tasks are allocated to resources (memory, network, CPU) as virtual machines. The concept of scheduling and resource allocation has different aim. We need to find a schedule to execute a DAG workflow on Infrastructure as a Service computing resource to minimize the execution cost and time.

Input: The schedule is defined in the format of S=(R, A, TC, TT) in the form of set of resources (R), task to resource allocation (A), total cost (TC) and total time (TT). For each resource R= (r_1, r_2... r_n) different virtual machines are allocated, and each resource has its start time ST_{ri} and end time ET_{ri}. Here A represents allocation and the allocation consist of set of tuples in the form of at a^r_t= (t_i, r_j, ST_{ri}, ET_{ri}). A task t_i is scheduled to run on the resourcer_j. The total cost TC and total time TT are calculated as follows:

$$TC = \sum_{i=1}^{R} C_{VM} \times \left\lfloor \frac{(ET_{r_i} - ST_{r_i})}{\tau} \right\rfloor \tag{1}$$

$$TT = \max \left\{ ET_{r_i} : t_i \varepsilon T \right\} \tag{2}$$

Output: Assignment of tasks to resources in minimized time and cost. The n number of tasks and n number of resources are allocated as {(t_1, r_1), (t_2, r_2)... (t_n, r_n)}.

Constraints: Each task must be completed within short time and without interruption. One virtual machine can complete one task at a time. The processing time is depends on the virtual machine allocated.

Objective: The aim is to assign each task to matching virtual machine resources and sequence the tasks to minimize the time, cost and throughput. The challenge of job scheduling and resource allocation was optimized by combing GABFO with minimized time and cost, finally reliability is achieved.

4. The standard bacterial foraging optimization

BFO algorithm was used to solve optimization problem [18]. Bacterial Foraging Optimization is an evolutionary method based on E.coli bacteria. The area having high level nutrients are searched by bacteria. This task is used for optimization process. By sending signals individual bacterium communicates with others. During foraging locomotion is achieved by a set of tensile flagella. E.coli bacteria tumble or swim using flagella. These are the two basic operation of bacteria performed at the time of foraging. After considering two previous factors foraging decision is taken by bacteria. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis. The basic idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.

Foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake E per unit time T spent foraging. Hence, they try to maximize a function like E/T. Maximization of such a function provides nutrient sources to survive and additional time for other important activities (e.g., fighting, fleeing, mating, reproducing, sleeping, or shelter building). Herbivores generally find food easily but must eat a lot of it. Carnivores generally find it difficult to locate food but do not have to eat as much since their food is of high energy value. The “environment” establishes the pattern of nutrients that are available and it places constraints on obtaining that food (e.g., small portions of food may be separated by large distances). During foraging there can be risks due to predators, the prey may be mobile so it must be chased and the physiological characteristics of the forager constrain its capabilities and ultimate success. Bacterial Foraging Algorithm is explained by following steps.

- Chemotaxis
- Swarming
- Reproduction and
- Eliminational-Dispersal

The chemotactic step was described by the equation

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^n \Delta(i)}} \tag{3}$$

Where $\Delta(i)^n$ is a n-dimensional randomly generated vector with elements within the following interval: [-1,1]. After that, each bacterium $\theta^i(j,k,l)$ modifies its position as mentioned in Eq. (4), where $C(i)$ is the stepsize for search direction $\varphi(i)$. Eq. (4) represents the swim of a bacterium

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(i) \tag{4}$$

5. The standard genetic algorithm

GA is an evolutionary algorithm introduced by Holland. GA mimics the process of natural evolution. It generates the solution to optimization problem using inheritance, crossover, selection and mutation.

- Produce initial population
- Evaluate fitness function
- Produce new population (using mutation and crossover)

The search in the genetic algorithm starts with an initial population. Each individual is evaluated by its fitness function. According to fitness value unfitted populations are eliminated. Individuals are manipulated using genetic operators. Totally three operators are used in genetic algorithm. At first the production operator is used to create copies of best fitted population. The low fitness values populations are eliminated. The second one is the crossover operator. This makes swapping of individual elements. Third one is the mutation operator. Application of this operator is used for random search.

6. The proposed hybrid GABFO algorithm

The objective of hybrid GABFO algorithm was get the minimum function $F(\varphi)$, $\varphi \epsilon R$, where φ represents the position of the bacteria. $F(\varphi)$ represents an attractant repellant profile and φ represents the position of the bacteria. The nutrients for the bacteria is located as, $F < 0, F = 0, F > 0$ specifies the presence and absence of nutrients. The natural area for the bacteria is represented as,

$$H(j,k,l) = \{\varphi^x(j,k,l) | x = 1,2,\dots,N\} \tag{5}$$

Eq. (5) shows the parameters of N bacteria at jth chemotactic step, kth reproduction step and lth elimination dispersal. Then $C(x,j,k,l)$ represent the cost of bacteria at ith position.

$$\varphi^x(i,j,k) \epsilon R^n \tag{6}$$

$$\varphi^x = (i+1,j,k) = \varphi^x(i,j,k) + C(x)\varphi(i) \tag{7}$$

The value of $C(i) > 0$ is the step size for each tumble. Here N_s is lifetime length of the bacteria calculated using chemotactic steps.

Algorithm: Hybrid GABFO Algorithm

```
Step 1: Initialize the input parameters $N$, $N_c$, $N_{re}$, $N_s$, $C(i)$
        $N$: No. of bacteria
        $N_c$: Chemotactic step
        $N_{re}$: Reproduction step
        $N_s$: No. of steps
        $C(i)$: Size of the step taken
Step 2: Calculate the elimination dispersal step using $j=j+1$

Step 3: Calculate reproduction step using $k=k+1$

Step 4: Calculate chemotaxis loop using $l=l+1$
            For $i=1, 2,...,N$ calculate fitness function FT ($i, j, k, l$)
            Let FT= FT ($i, j, k, l$). Save this value to find better cost.
            Tumble: generate a random vector on [-1, 1]
            Move: When
              $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta(i)^n \Delta(i)}}$
            Compute FT($i, j+1, k, l$)
            Perform swim
            Else, Go to next bacterium ($i$,1)
Step 5: If $j<N_c$, go to step 3. Bacterium life is not over.
```

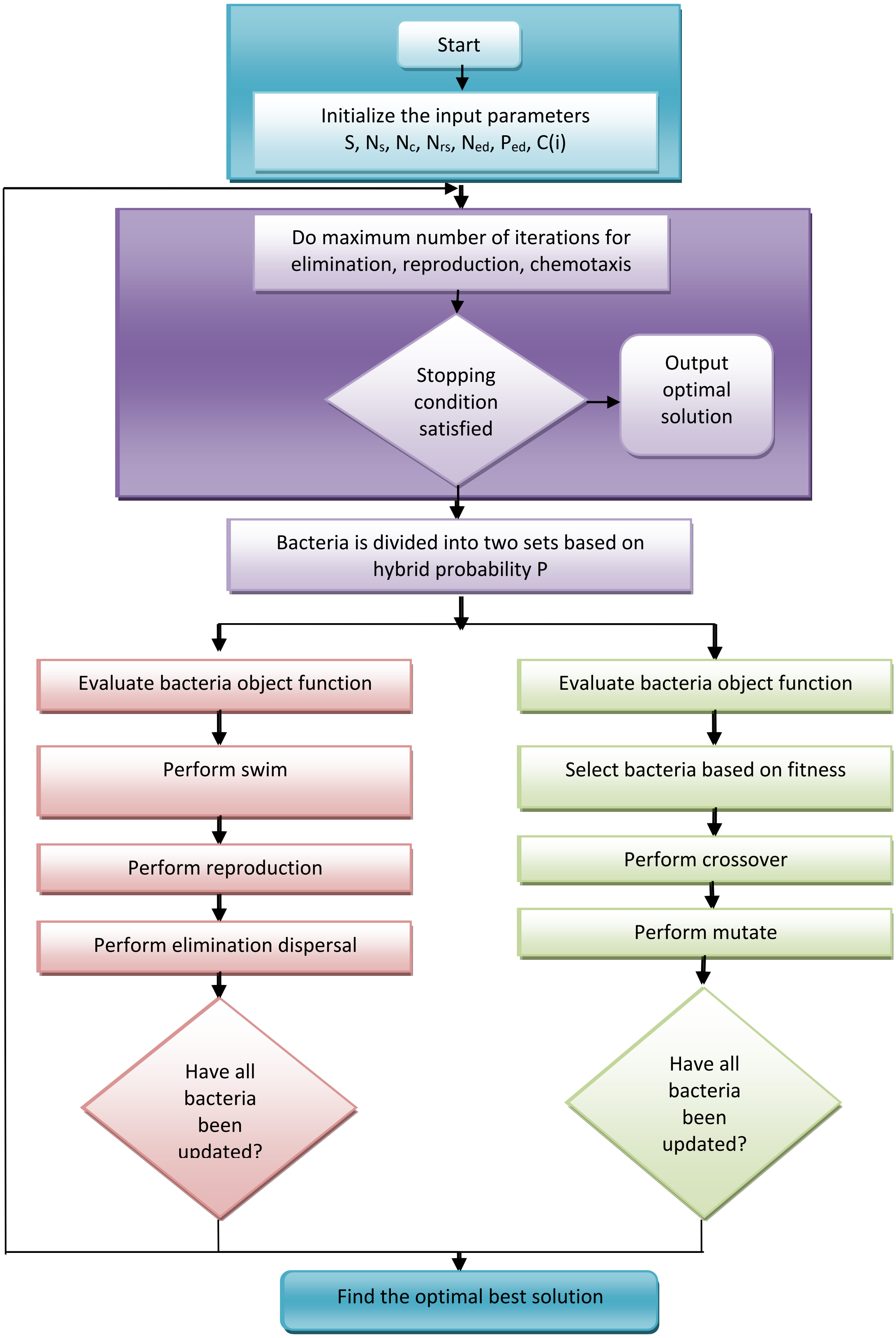
Start
Initialize the input parameters
S, N_s, N_c, N_{rs}, N_{ed}, P_{ed}, C(i)
Do maximum number of iterations for elimination, reproduction, chemotaxis
Stopping condition satisfied
Output optimal solution
Bacteria is divided into two sets based on hybrid probability P
Evaluate bacteria object function
Perform swim
Perform reproduction
Perform elimination dispersal
Have all bacteria been updated?
Evaluate bacteria object function
Select bacteria based on fitness
Perform crossover
Perform mutate
Have all bacteria been updated?
Find the optimal best solution

Figure.1 Flowchart for GABFO

Step 6: Calculate Reproduction step

$$FT = \sum_{j=1}^{Nc+1} FT(i,j,k,l)$$

Step 7: Elimination dispersal: Eliminate and disperse bacterium with probability P_{ed}.

If $l<N_{ed}$ then go to step 2,
Otherwise end.

7. Simulation and analytical results

For simulation we used CloudSim [3] as a tool for testing and analyzing new algorithms for creation and allocation of virtual machines to Cloudlets for execution. This CloudSim is used to create a simulation environment for cloud. In the proposed work we considered two physical machines (PM) and we created 4 virtual machines (VM1, VM2, VM3, VM4) from 2PM for testing. Here cloudlets are the jobs or tasks in the simulation environment; each cloudlet is assigned to individual virtual machine. We are submitting 10 types of jobs as workflow model to the physical machine. The given jobs are divided into number of 0-50 tasks. Totally 52 iterations are calculated for each run. And 10 times the workflow is executed under different loads to get different execution results. Following table 1 shows the simulation parameters considered under our setup. The output parameters we considered over five algorithms are execution time, cost and throughput. Our result gives the reduced time and cost as well as increased throughput for various workloads. Compared to PSO, GA and BFO the hybrid GABFO produced better result. The work mentioned in [24] compared with our result.

Fig.2 shows the result of total execution time of jobs. Increases in number of instances directly affect the execution time adversely. But, the proper virtual machine placement and sharing the jobs during the demand situation reduces the time consumption effectively. The proper VM selection and the prior load consumption are the major requirements for minimum execution time. The total cost is defined as the cost of resource and the amount of total time period the resource used. The results of cost versus various jobs are shown in Fig.3. More amounts of jobs completed in a minimal duration are called as throughput. Higher throughput gives the better result. If the throughput increases resource utilization also increases. Throughput result is shown in Fig.4. Resource utilization is defined in Fig.5 shows that, the number of allotted resources divided by the total number of available resources. When the number of jobs varied resource utilization also varied. From the analysis the execution time was reduced upto 23.67% and the cost was reduced upto 13.05%.

Table 1. Simulation parameters

Resource Parameter	Quantity
Number of PM	2
Number of VMs	4
Number of job	1-10
Number of task	0-50
Number of iteration	52
Number of executions	10

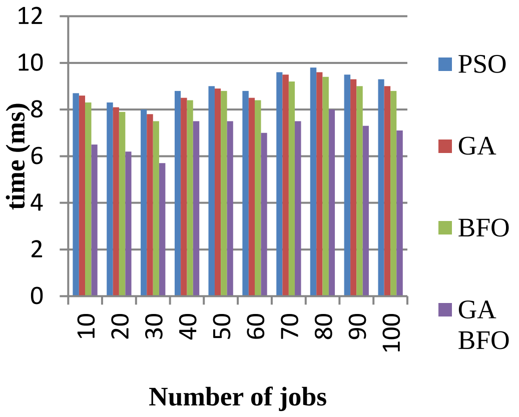

Figure. 2 Total time

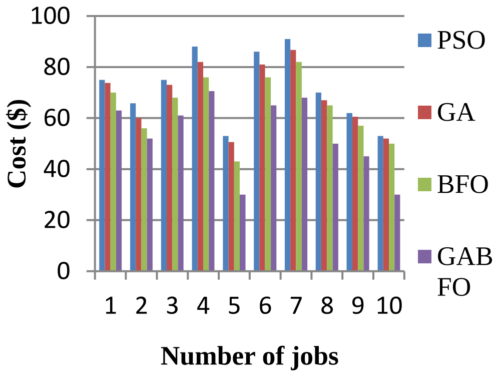

Figure. 3 Cost

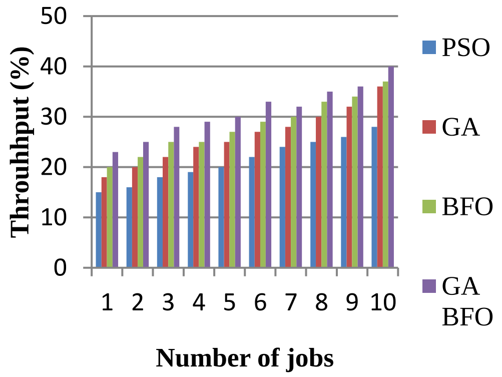

Figure. 4 Throughput

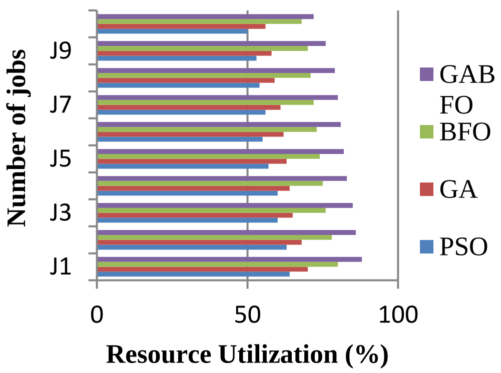

Figure. 5 Resource utilization

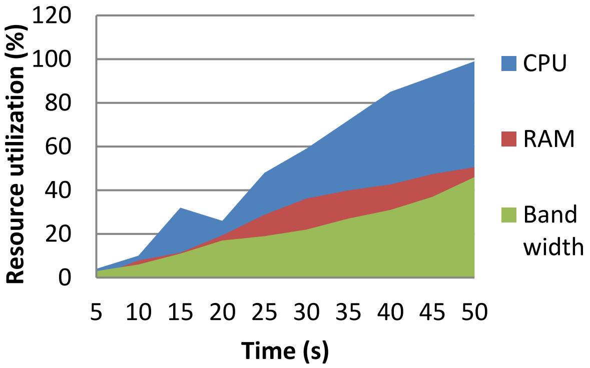

Figure. 6 Virtual machine 1

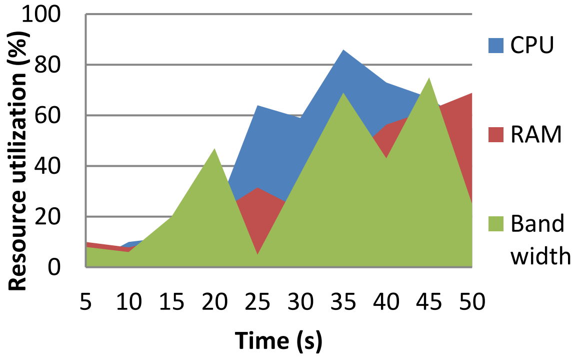

Figure. 7 Virtual machine 2

The maximum utilization of resources is important in cloud. Because better resource utilization gives better performance. Our test results produce maximum utilization of resources compared to previous works. The above Figs. 6 and 7 show the sample results of resource utilization under virtual machine 1, virtual machine 2. These two virtual machines are placed under single physical machine. CPU, RAM and bandwidth are the resources we considered for allocation and utilization. Above figure shows that approximately the resources are utilized over 91.53%.

8. Conclusion and future work

In this work, we addressed the "job scheduling and resource allocation problem", which is to schedule and allocate virtual resources to achieve high resource utilization to meet user's needs with minimum utilization parameters. The proposed work provided the solution to the scheduling and resource allocation problems using optimized hybrid algorithms. Variations in genetic algorithm have been investigated and implemented for learning and improve the speed of convergence. This work proposed a novel hybrid approach consisting of Genetic Algorithm (GA) and Bacterial Foraging Optimization Algorithm (BFOA) also the performance of scheduling and resource allocation was tested with varying execution steps. These proposed experimental problems were tested with CloudSim tool. Compared to PSO, GA, BFO algorithms the hybrid GABFO demonstrated the better scheduling results within better cost saving scheme. Our simulation result shows that the resources are highly utilized with optimized performance result. We can apply this algorithm to set a real cloud environment.

One of the extension of this work plans to execute the defined various workflow models with this same algorithm in simulated environment. It would be interesting to investigate the performance of scheduling workflows or allocating resources to support various workflow models. Second plan is to extend the work by automatic resource allocation system using self learning algorithm. Which means each problem and their solutions are learned by the automatic decision system. If the same job appears again without any execution the resources are automatically allocated.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", *National Institute of Standards and Technology Special Publication, SP 800-145*, p.7, 2011.

[2] D. Rani and R.K. Ranjan, "A Comparative Study of SaaS, PaaS, and IaaS in Cloud Computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.4, No.6, pp.158-161, 2014.

[3] F. Zhang, J. Cao, K. Li, S.U. Khan, and K. Hwang, "Multi-objective scheduling of many tasks in cloud platforms", *Elsevier-Future Generation Computer Systems*, Vol. 37, pp. 309–320, 2014.

[4] J. Yu and R. Buyya, "Workflow scheduling algorithms for grid computing, Metaheuristics for Scheduling in Distributed Computing Environments", *Springer*, Germany, Vol. 146, pp. 173, 2008.

[5] M. Mezmaz, N. Melab, Y. Kessaci, Y.C. Lee, E.G. Talbi, A.Y. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems", *Journal of Parallel and Distributed Computing, Elsevier*, vol. 71, No. 11, pp. 1497-1508, 2011.

[6] G. Malathy, R. Somasundaram, and G. Vidhya, "A survey on workflow scheduling algorithms and map reduce method in cloud environment", *International Journal of Communications and Engineering,* Vol. 6, No. 6, pp. 1-6, 2012.

[7] J. Gasior and F. Seredynski, "A decentralized multi-agent approach to job scheduling in cloud environment", *Intelligent Systems, Spring*er, Vol. 322, pp. 403-414, 2015.

[8] N. Anitha and A. Basu, "A dynamic resource allocation based on multi attributes scoring in collaborative cloud computing", *Global Journal of Computer Science and Technology*, Vol. 15, No. 4, pp. 1-7, 2015.

[9] Z. Xaio, W. Song, and W. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment", *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, pp.1107-1117, 2013.

[10] A. Bamini and S. Enoch, "Role of Cloud Computing in the Provision of Healthcare", *World Journal of Medical Education and Research*, Vol.2, No. 1, pp. 12-17, 2012.

[11] M. Abdullah and M. Othman, "An improved genetic algorithm for job scheduling in cloud computing environment", *Procedia Information Technology and Computer Science*, Vol.2, pp. 291-296, 2013.

[12] J. Liu, X.G. Luo, X.M. Zhang, F. Zhang, and B.N. Li, "Job scheduling model for cloud computing based on multi-objective genetic algorithm", *International Journal of Computer Science Issues*, Vol. 10, No. 1, pp. 134-139, 2013.

[13] A. Bamini and S. Enoch, "Optimized Resource Scheduling Using Classification and Regression Tree and Modified Bacterial Foraging Optimization Algorithm", *International Journal of Applied Engineering Research*, vol. 10, No. 16, pp. 37170-37175, 2015.

[14] S. Kaur and A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment", *International Journal of Information Technology and Computer Science*, Vol. 10, pp. 74-79, 2012.

[15] Y. Ge and G. Wei, "GA-Based task scheduler for cloud computing systems", *IEEE International Conference on Web Information Systems and Mining*, pp. 181-186, 2010.

[16] M. Shojafar, S. Javanmardi, S. Abolfazli, and N. Cordeschi, "FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method", *Cluster Computing*, Vol. 18, No. 2, pp. 829-844, 2015.

[17] F. Pop, R.I. Tutueanu, C. Barbieru, M.A. Vasile, and J. Kolodziej, "Adaptive resource allocation in cloud computing based on agreement protocols", *Intelligent Agents in Data-intensive Computing, Springer,* Vol. 14, pp. 193-213, 2016.

[18] S. Zhan and H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing", *Journal of Information and Computational Science*, Vol. 9, No. 13, pp. 3821-3829, 2012.

[19] A. Shokripour, M. Othman, H. Ibrahim, and S. Subramanian, "New method for scheduling heterogeneous multi-installment systems", *Future Generation Computer Systems*, Vol. 28, pp. 1205-1216, 2012.

[20] A.K. Bardsiri and S.M. Hashemi, "A review of workflow scheduling in cloud computing environment", *International Journal of Computer Science and Management Research*, Vol. 1, No.3, pp. 1-4, 2012.

[21] W. Wang, "Bayesian cognitive model in scheduling algorithms for data intensive computing", *Journal of Grid Computing*, Vol.10, pp. 173-184, 2012.

[22] G. Malathy, R. Somasundaram, and G. Vidhya, "A survey on workflow scheduling algorithms and map reduce method in cloud environment", *International Journal of Communications and Engineering*, Vol. 6, No. 6, pp. 1-6. 2012.

[23] J.O. Gutierrez-Garcia and K.M. Sim, "A family of heuristics for agent-based elastic cloud bag-of-tasks concurrent scheduling", *Future Generation Computer Systems*, *Elsevier,* vol. 29, No. 7, pp. 1-18, 2012.

[24] N. Kumar and S. Saxena, "A preference-based resource allocation in cloud computing systems", *Proceedia Computer Science*, Vol. 57, pp. 104-111, 2015.

[25] B. Xu, C. Zhao, B. Hu, and E. Hu, "Job scheduling algorithm based on berger model in cloud environment", *Advances in Engineering Software*, Vol. 42, No. 7, pp. 419-425, 2011.

[26] S.S. Kim, J.H. Byeon, H. Yu, and H. Liu, "Biogeography based optimization for optimal job scheduling in cloud computing", *Applied Mathematics and Computation*, Vol. 247, pp. 291-296, 2014.

[27] S. Yeo and H.H.S. Lee, "Using mathematical modeling in provisioning a heterogeneous cloud computing environment", *Computer*, Vol. 44, No. 8, pp. 55-62, 2011.

[28] R. Mohana, "A position balanced parallel particle swarm optimization method for resource allocation in cloud", *Indian Journal of Science and Technology*, Vol. 8, No. 3, pp. 182-188, 2015.

[29] Z. Yang, X. Qin, W. Li, and Y. Yang, "Optimized task scheduling and resource allocation in cloud computing using PSO based fitness function", *Information Technology Journal*, Vol. 12, No. 23, pp. 7090-7095, 2013.

[30] Z. Yang, X. Qin, W. Li, and Y. Yang, "Optimized task scheduling and resource allocation in cloud computing using PSO based fitness function", *Information Technology Journal*, Vol. 12, No. 23, pp. 7090-7095, 2013.

[31] R. Mohana, "A position balanced parallel particle swarm optimization method for resource allocation in cloud", *Indian Journal of Science and Technology*, Vol. 8, No. 3, pp. 182-188, 2015.

[32] K. Chandran, V. Shanmugasundaram, and K. Subramani, "Designing a fuzzy logic based trust and reputation model for secure resource allocation in cloud computing", *International Arab Journal of Information Technology*, Vol. 13, No. 1, pp. 30-37, 2016.

[33] S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham, "Hybrid job scheduling heuristics for cloud computing environment", *Advances in Intelligent Systems and Computing, Springer*, Vol. 303, pp. 43-52, 2014.

[34] A. Bamini and S. Enoch, "Dynamic Scheduling and Resource Allocation in Cloud", *International Journal of Control Theory and Applications*, Vol. 10, No. 3, pp. 63-72, 2017.

[35] G.K. Shyam and S.S. Manvi, "Resource allocation in cloud computing using agents", In: *Proc. of the IEEE International Advance Computing Conference*, pp. 458-463, 2015.

[36] Z. Zhigang, Z. Hongli, Y. Xiangzhan, and G. Junwu, "Continuous resource allocation in cloud computing", In: *Proc. of the IEEE International Conference on Communications*, pp. 319-324, 2015.