TASK SCHEDULING MECHANISM FOR CLOUD COMPUTING USING HYBRID PARTICLE SWARM OPTIMIZATION ALGORITHM

A Daniel[1] Dr.N.Krishnaraj[2]
[1]Research Scholar, Shri Venkateshwara University, Gajraula, Uttar Pradesh 244236, India
Asst.Prof, School of Computing Science & Engineering, Galgotias University,Greater Noida, Uttar Pradesh 226001, India
[2]Professor,Department of Computer Science and Engineering, Sasi Institute of Technology and Engineering, Tadeplalligudem 534101,Andrapradesh,India

Abstract

As the world is advancing towards progressively proficient computing and quicker methodologies, cloud computing is a well known computing model to such expanding necessities. So as to give financially executions in cloud condition, task scheduling process is highly crucial. This paper proposes a method by combining the genetic algorithm (GA) and the particle swarm optimization (PSO) algorithm to improve load balancing and cloud throughput. We assess the results utilizing the CloudSim toolbox and reproduction results exhibit that the proposed methodology has a superior act regarding makespan, improvement proportion, and so on.

Keywords: Task scheduling; Particle swarm optimization; Genetic algorithm; Cloud computing

1. Introduction

Logical computing attempts to beat the substantial scale issues in various fields, for example, Earth Sciences, High-Energy Physics, and Bioinformatics. These controls dependably need enormous measures of assets to work required tasks and trials, for example, parameter clear tests (PSEs) [1]. Cloud computing is a prominent stage which suits well in comprehending elite computing issues by interfacing a substantial number of frameworks through a system. Cloud computing in this setting portrays a worldview to convey the computational assets to buyers as an open administration as per the compensation scrutinizes show. Suitable assignment planning is important to give a high productivity in cloud condition. Undertaking planning for the disseminated frameworks is a NP complete issue, so customary booking techniques don't give sensible productivity in such conditions [2].

Undertaking planning is a key worry for appropriated situations that must deal with a few errands in numerous assets, while improving asset use and the makespan. For the most part, task booking system maps submitting assignments to accessible assets in cloud condition as indicated by their qualities to accomplish superior computing. Errand booking process in cloud framework can be abridged to the accompanying advances [3]:

- Resource finding: Broker finds every single accessible asset in the framework and stores all related data, for example, the limit, preparing cost, and the present load of assets.

- Resource choice: The reasonable asset is picked dependent on the errand necessities and asset qualities.
- Task accommodation: The errand is relegated to the asset that is picked

As appeared in Fig. 1, the scheduler for example Datacentre dealer endeavors to delineate to the appropriate virtual machines for decreasing makespan, exchanging time, costs, and so forth. By and large, booking algorithm can be isolated into ideal and imperfect dependent on the nature of arrangement. In ideal booking system, the ideal employment asset mappings are resolved by the total data about condition status, for example, load and capacities of equipment. In the event that fundamental data isn't accessible or there isn't sufficient opportunity to locate the ideal arrangement, imperfect techniques are connected rather.

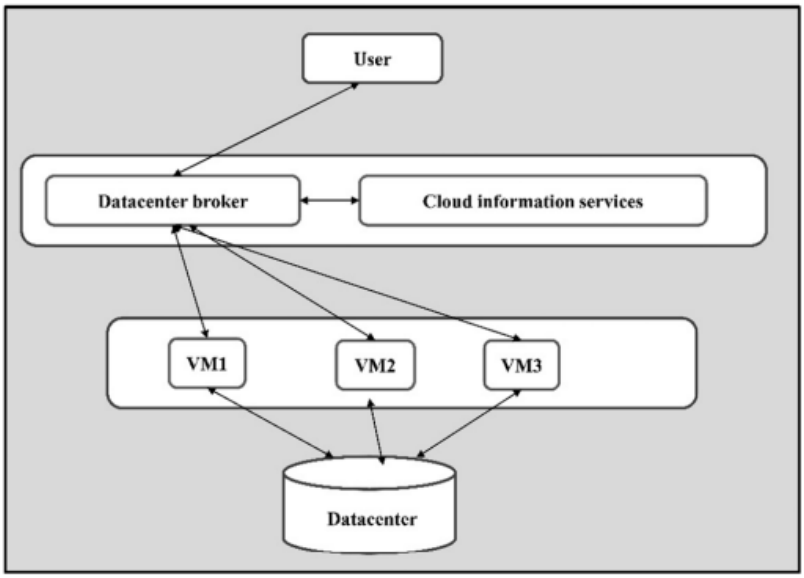

Fig. 1. Task scheduling process in cloud environment.

Problematic strategies are additionally sorted into heuristic and surmised. Heuristic algorithms make as couple of suspicions as workable for status of assets, for example, load or about length of occupation before employment planning. Surmised planning techniques depend on a similar info data and formal computational model as ideal booking strategies yet they beat the NP-fulfillment of ideal schedulers by lessening the arrangement space. Clearly gathering all information gives new issues practically speaking [4]. In this way, heuristic systems are normal and valuable. Swarm knowledge approaches are exceptionally prevalent and compelling in dispersed planning. The principle reason is that they can illuminate advancement issues without need of an excess of heretofore data about the issue.

2. Related works

Undertaking scheduling is a primary worry in cloud condition, such huge numbers of research worked in this extension to improve proficiency. The issue of finding an ideal timetable for a lot of errands is NP-hard. Up until this point, no brisk arrangements have been found for this issue, and may not be found later on by any stretch of the imagination. By and large, methods that discover arrangement dependent on the full scan are

not appropriate for this sort of issues because of the expense of procedure is exceptionally high. Metaheuristic algorithms can defeat this issue by furnishing an adequately decent arrangement particularly with fragmented information or restricted calculation limit. For instance, ACO system is helpful for taking care of independent optimization issues which can be decreased to deciding great ways through diagrams. ACO has been connected to tackle static, dynamic, discrete and issues, for example, sales rep issue, job shop scheduling, and job scheduling in lattice and cloud computing, and considerably more [5].

[6] proposed a joined scheduling named FUGE that depends on fuzzy inference and genetic algorithm (GA) to improve execution time. FUGE system considers fuzzy hypothesis in wellness and hybrid advances. It indicates jobs as qualities and allots calculation components to these qualities. FUGE system performs with two chromosome types. The principal type thinks about length of job, speed of CPU, and size of RAM. The second sort thinks about length of job and asset bandwidth. These parameters are the contribution of fuzzy inference framework. FUGE figures the wellness esteem by fuzzy capacity for every chromosome of each kind. At that point, it performs hybrid on two candidate chromosomes and makes new chromosome. At long last, the chromosome with the biggest wellness esteem is included new populace. The CloudSim results showed that FUGE technique could successfully diminish execution time and execution cost contrasted with GA, MGA [7]), ACO, and MACO.

[8] introduced another errand scheduling called IGATS dependent on improved genetic algorithm in cloud condition. The principle commitment of IGATS algorithm is load need definition dependent on the dynamic qualities of the cloud framework. It considers number of jobs in the line in a given timeframe for first need load factor. At that point it considers memory and CPU usage as second need load factors. In the continuation, it accept bandwidth usage as third factor. Reenactment tests demonstrated that IGATS could significantly improve cloud throughput and execution time.

[9]) displayed a dynamic job scheduling algorithm by utilizing Modified PSO technique in cloud to limit job execution time. Standard PSO procedure randomly creates essential particles, so the likelihood of uniting to the best arrangement is diminished. In opposite, the proposed Modified PSO methodology consolidates shortest job to fastest processor method (SJFP) into PSO for making beginning populace. Thusly, in the event that there are m jobs and n accessible virtual machines, at that point particles ought to be spoken to in network m × n. Every single other stage are indistinguishable to standard PSO system. The creators assessed Modified PSO scheduling methodology in Eucalyptus cloud which is created on IPCRC focal point of Amirkabir University. The outcomes exhibited that changed PSO algorithm could decrease finishing time of job execution in examination with the PSO and Genetic algorithms.

[10] proposed another load and blame mindful scheduling dependent on the Honey Bee algorithm to upgrade QoS of cloud condition. The honey bee scheduling algorithm indicates datacenters as scout bees. At that point, it initiates scout bees for picked hubs and ascertains estimation of qualification for datacenter and

picks the fittest bee from each datacenter. The proposed procedure considers blame rate, arrange load, framework load, and commencement time in wellness work. Relative examination in CloudSim showed the sensible execution of - c in blame mindful condition

[11] proposed another job scheduling system dependent on the Ant Colony Optimization algorithm to decrease normal makespan. The proposed methodology lands all data of positions and VMs and evaluations the Expected Time to Compute (ETC). At that point, it introduces important parameters, for example, pheromone vanishing rate, number of ants, pheromone preliminary esteem, and so on. Moreover, it ascertains the likelihood for choosing a VM for an errand and chooses a VM with the most astounding likelihood. At last it processes the makespan of calendar worked by every ant and chooses the timetable with the least makespan. The logical outcomes showed that ACO procedure could decrease total execution time more than FCFS method regardless of number of jobs.

[12] improved execution of ACO scheduling procedure by utilizing different methods, for example, Particle Swarm Optimization (PSO) in cloud. The proposed procedure utilizes ACO system to decide a few arrangement sets dependent on the refreshed pheromone, and then applies PSO technique to locate the best arrangement. Exploratory outcomes showed that half and half scheduling technique improved the combination speed. [13] introduced dynamic undertaking scheduling dependent on Bat procedure to improve by and large execution of cloud condition. Bats assess the separation of their prey by echolocation. They fly randomly to find prey with speed, uproar, and recurrence. After they accomplish their sustenance, they change their din, recurrence and heartbeat rate of discharge as per the separation of them and the nourishment. [14] presented a BAT-Gravitational half breed scheduling algorithm dependent on the due date limitations and trust display. Gravitational system thinks about arrangements as articles and measures their quality by their masses. Gravity constrain moves articles to different items with heavier masses. What's more, the overwhelming masses move more gradually than lighter masses, so abuse of the system comparing to the best arrangement is given. The BAT-Gravitational procedure characterizes correspondence trust as accessible bandwidth between server farm and client. At that point it chooses assets for jobs as indicated by their trust esteem. Experimental results with CloudSim toolkit demonstrated that BAT-Gravitational scheduling procedure could lessen time execution 8.68% contrasted with random scheduling technique.

[15] presented crossover PSO-MOBA (Particle Swarm Optimization and Multi-Objective Bat Algorithm) in cloud condition. Nearby space looking is done through PSO system and MOBA plays out the worldwide refreshing. It applies M/M/m lining model to control various assets and jobs dependent on the charge of administration, execution time, and business cost. Specialist organization doles out jobs so that total benefit and use of assets are augmented. In this manner, half breed PSO-MOBA system demonstrates the advantages of both PSO and MOBA methods and accomplishes quicker intermingling.

3. Proposed Method

3.1. Hybrid Particle Swarm Optimization Algorithm

This segment exhibited a hybrid parallel algorithm by joining the genetic algorithm and molecule swarm algorithm hybrid (HGAPSO-MD). In the atomic elements reproduction computing framework, the execution of parallel algorithm has further improvement than past algorithms.

3.2. Algorithm design idea

Despite the fact that the incredible particular weight can influence the union rate in the GA choice task process, people whose wellness esteems are not useful to taking care of the issue will be dispensed with rapidly. Along these lines, populace decent variety will be harmed. In spite of the fact that new people will be delivered by means of crossover and mutation operations, its check is too low. In spite of the fact that the decrease of particular weight can advance the likelihood of algorithms to look through the worldwide ideal, it will diminish its assembly rate. So as to hold great execution, molecule swarm optimization can be added to the genetic algorithm usage process. This hybrid algorithm is overwhelmed by GA and enhanced by PSO. The GA can be utilized to lead a worldwide inquiry in the underlying task time frame. The PSO can be connected to speed up the combination rate in the later period dependent on the decreased size of populace. Along these lines, people in the last age after ordinarily of cycles will draw nearer to the worldwide ideal arrangement with a moderately quick assembly rate. The particular presentation is as per the following.

3.2.1. Fitness function

Wellness is utilized to gauge the level of brilliance of people in the populace that might almost certainly accomplish or help to locate the ideal arrangement. In this way, its choice is very significant. The work process plan means to bring out little makespan and flowtime. The paper utilizes flowtime/m to assess positive or negative algorithms, of which m is the quantity of legitimate hubs..

3.2.2. Crossover

Many properties such as average fitness and diversity are changing in the evolutionary process. Therefore, a fixed parameter can hardly adapt to the whole process of scheduling algorithms. This algorithm adopts dynamic crossover operators. The population diversity is damaged by the selection operation in the early period. Thus, new and optimal individuals shall be searched with great effort. The crossover probability in the early period $p_c(range\ of\ variation: 0.4\ 0.9)$ is quite large. However, individuals have basically focused on areas near the optimal solution space in the later evolution process. At this time, some good discovered genes shall be kept to reduce p_c. This algorithm sets a threshold value t, which is used to calculate the average fitness of progeny in the crossover process. Further, it is used to set the minimum average fitness of progeny populations. If it is continued for t generations, the average fitness of progeny populations will be less than the minimum average fitness. Then, p_c is updated as a relatively large value between $p_c \times 90\%$ and p_{cmin}, of which p_{cmin} is the minimum crossover probability.

3.2.3. Mutation

Similar to crossover operators, this paper adopts dynamic mutation operators. The mutation probability $p_m(range\ of\ variation: 0.01\ 0.1)$ can be downsized appropriately based on rich genetic forms of individuals in the population instead of convergent algorithms. However, when the algorithm gets close to or has been converged, pm can be enlarged appropriately based on a constant or slightly changed population evolution rate. This algorithm sets a threshold value to calculate the evolution rate of progeny populations in the mutation process. Further, it is used to set the minimum evolution rate of progeny populations. In the mutation process, if it is continued for t generations, the evolution rate of progeny populations is smaller than the minimum evolution rate.p_v is updated as a small value between $p_v \times 110\%$ and p_{mmax} , of which p_{mmax} is the maximum mutation probability [53].

3.2.4. Switching point setting of hybrid algorithm

In order to give full play to the global convergence ability of GA as well as rapid convergence of PSO, seeking an optimum point has become the key to the hybrid algorithm. This paper adopts the following dynamic fusion strategies to guarantee fusion between GA and PSO at the optimal timing.

(1) set a switch coefficient $S \in (0\ 1)$; then determine its value after many trials. In order to avoid it being caught in an infinite loop or beyond system loading, this paper sets conditions to judge whether the algorithm shall be terminated; that is, the maximum time of calling termination evaluation function is R. Then, the maximum number of evaluation functions of GA is $T_{max} = R \times S$;

(2) Set a threshold value L, record the number of times the maximum mutation probability $pmmax$ is used in the mutation operation of GA. If the number of times the maximum mutation probability has been used in the mutation process has exceeded L, this indicates a too low evolution rate of progeny populations; at this time, the optimization speed of GA is very low. The genetic algorithm process can be terminated. Record iterations T;

(3) If a satisfactory solution is not found yet without exceeding the given T_{max}, it has been in a stable state; if the fitness cannot be further improved, the algorithm can be terminated. That is, the practical calls of GA shall be (T_{max}, T)

3.2.5. Adaptive parameter ω of HGAPSO algorithm

The population can reduce gradually the search space by adjusting adaptively the parameter ω as the iterations increasing. So the adaptive PSO $(APSO)$ algorithm is more effective, because the parameter ω here also reduces step by step , so the search space shrinks step by step. An excellent adaptive inertia weight factor ω was designed and it can keep nice balance between convergence and population diversity in the $HGAPSO$ algorithm. ω is formulated as follows:

$$\omega = (\omega_{max} - \omega_{min}) * \exp\left(\frac{g_{max} - g}{g_{max}}\right) - \omega_{min} \quad (1)$$

4. Results and discussion

This part discuss about the experimental studies that are assumed with a view to focus on the projected method efficiency. The proposed method is compared with some existing method in terms of degree of imbalance (DI), makespan and total execution time with the varying number of tasks.

Degree of Imbalance (DI)

It denotes the imbalance between virtual machines and expressed as

$$T_i = \frac{Total\_TL_j}{Num\_Pe_j * MIPS\_Pe_j} \quad (2)$$

Let $Total\_TL_j$ demonstrates sum of tasks that are allotted to the VM_j, Num_Pe_j refers sum of processing components of $MIPS\_Pe_j$ and VM_j shows MIPS of processing element of VM_j.

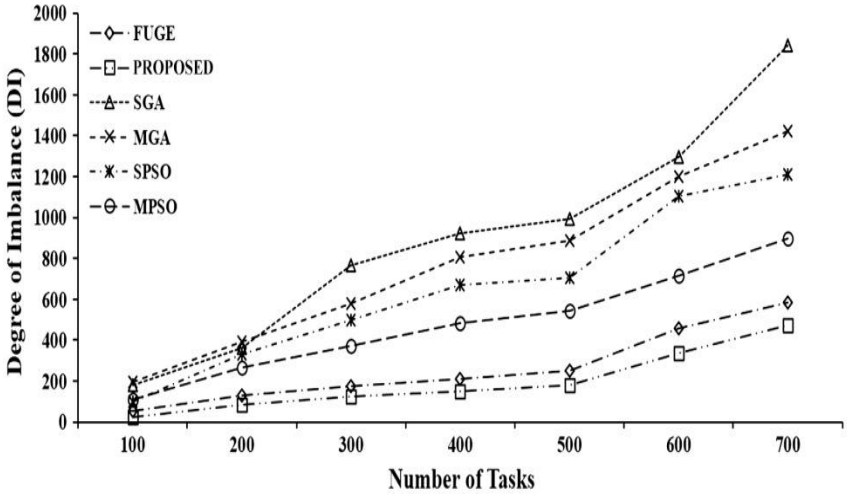

Fig. 2. Imbalance analysis

Fig. 2 shows the comparison of various methods in terms of DI with the varying number of tasks. When 100 number of tasks are given, all the method shows little variances. The variance increases gradually when the number of tasks get increased. When compared with all the other methods, FUGE attains low DI but does not shows enhanced results than the projected method. Every task should be spread evenly among the VMs. The proposed strategy reduces DI by 19%, 74%, 65%, 59% and 46% in comparison with FUGE, MGA, SGA, SPSO and MPSO correspondingly. Even though in increasing task counts, the proposed methodology exhibits enhanced results.

Makespan

The general condition which show the completing time over the entire tasks. When there is lowest makespan rate, the task scheduling method is efficient in assigning those to VM. It is expressed as

$$Makespan = \max\{FT_i | \forall i \in list\ of\ tasks\} \quad (3)$$

where FT_i denotes the completing time of task i. The average makespan for various scheduling methods with different number of tasks are given in Fig. 3. When comparing with SPSO strategy, FUGE strategy comprise

low makespan. With lower length tasks, the FUGE doesnot waste energy. The makespan attained by the projected method is decreased by 13%. The major cause is that the projected strategy assumes the characteristics of tasks, total execution time. For instance, while the cloud tasks are 700, the variance among ms among SPSO and FUGE strategy is over 674, and among FUGE and proposed method is 343.

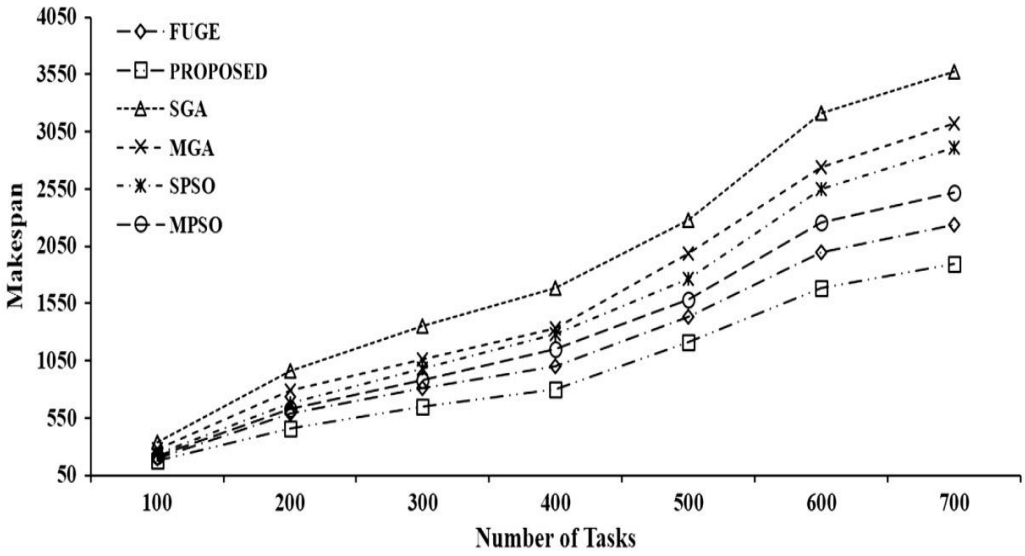

Fig. 3. Makespan for different number of tasks

Total execution time

Total execution time is estimated by

$$Texec = \sum_{j=1}^{m} exec_j \tag{4}$$

where m refers VMs counts and $exec_j$ denotes execution time for tasks that are assigned to VM_j. Total execution time is given in Fig. 4 with varying number of tasks.

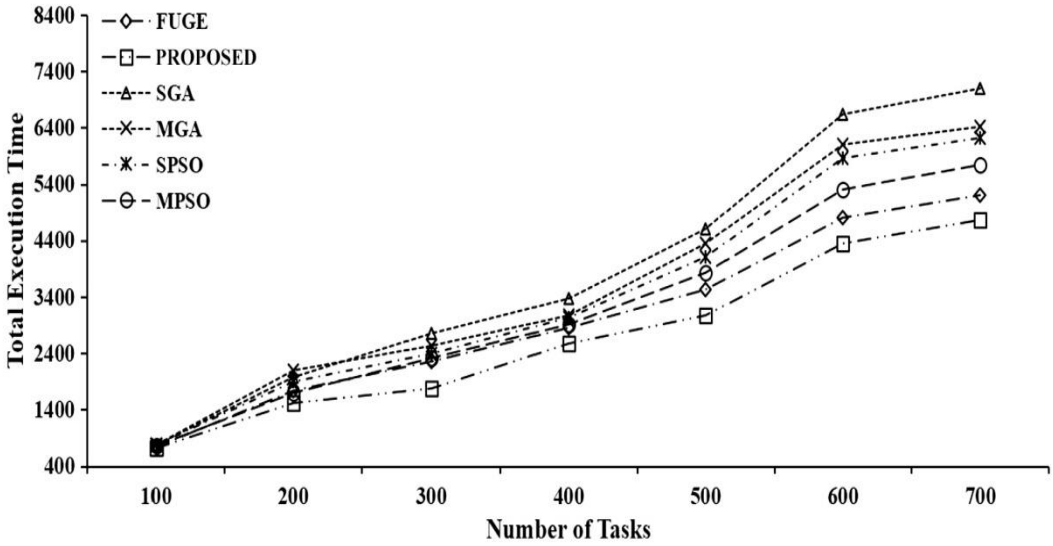

Fig. 4. Total execution time for different number of tasks.

The resources are adequate while the number of tasks is small. When comparing with projected method SPSO strategy takes high execution time. Particle move around the optimal position in SPSO algorithm with

discover the entire search space. Weak exploration and prematurity are the two major disadvantage of SPSO. The issues were overcome by the projected method. When comparing with all the method, the projected method gives decreases the total execution time by 8 to 16% on an average. The proposed method finds various solutions are enhanced the capacity of global search.

5. Conclusion

Cloud computing is a well known computing model to such expanding necessities. So as to give financially executions in cloud condition, task scheduling process is highly crucial. This paper proposes a method by combining the genetic algorithm (GA) and the particle swarm optimization (PSO) algorithm to improve load balancing and cloud throughput. We assess the results utilizing the CloudSim toolbox and reproduction results exhibit that the proposed methodology has a superior act regarding makespan, improvement proportion, and so on. When comparing with all the method, the projected method gives decreases the total execution time by 8 to 16% on an average. The proposed method finds various solution are and enhanced the capacity of global search.

References

[1] Elhoseny, M., Shankar, K., Lakshmanaprabu, S.K., Maseleno, A. and Arunkumar, N., 2018. Hybrid optimization with cryptography encryption for medical image security in Internet of Things. *Neural Computing and Applications*, pp.1-15.

[2] Karthikeyan, K., Sunder, R., Shankar, K., Lakshmanaprabu, S.K., Vijayakumar, V., Elhoseny, M. and Manogaran, G., 2018. Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA). *The Journal of Supercomputing*, pp.1-17.

[3] Shankar, K., Elhoseny, M., Chelvi, E.D., Lakshmanaprabu, S.K. and Wu, W., 2018. An Efficient Optimal Key Based Chaos Function for Medical Image Security. *IEEE Access*, *6*, pp.77145-77154.

[4] Pacini, E., Mateos, C., & García Garino, C. (2014). Distributed job scheduling based on Swarm Intelligence: A survey. Computers and Electrical Engineering. 40, 252–269.

[5] Mahato, D. P., Singh, R. S., Tripathi, A. K., & Maurya, A. K. (2017). On scheduling transactions in a grid processing system considering load through Ant Colony Optimization. Applied Soft Computing, 61, 875–891

[6] Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint metaheuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, 18, 829–844.

[7] Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science, 10, 74–79

[8] Ma, J., Li, W., Fu, T., Yan, L., & Hu, G. (2016). A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing. Wireless Communications Networking and Applications, 829–835.

[9] Abdi, S., & Sharifian, S. (2014). Task scheduling using modified PSO algorithm in cloud computing environment. International Conference on Machine Learning, Electrical and Mechanical Engineering (ICMLEME) (pp. 37–41).

[10] Gupta, P., & Ghrera, S. P. (2014). Load and fault aware Honey Bee scheduling algorithm for cloud infrastructure. Proceedings of the 3rd international conference on frontiers of intelligent computing: Theory and applications (pp. 135–143). .

[11] Selvaraj, S., & Jaquline, J. (2016). Ant Colony optimization algorithm for scheduling cloud tasks. International Journal of Computer Technology & Applications. 7(3), 491–494

[12] Wen, X., Huang, M., & Shi, J. (2012). Study on resources scheduling based on ACO algorithm and PSO algorithm in cloud computing. 11th international symposium on distributed computing and applications to business, engineering & science (pp. 219–222)

[13] Jacob, L. (2014). Bat algorithm for resource scheduling in cloud computing. International journal for research in applied science and engineering technology. 2, 53–57

[14] Kumar, V. S., & Aramudhan, D. (2014). Hybrid optimized list scheduling and trust based resource selection in cloud computing. Journal of Theoretical and Applied Information Technology, 69(3), 434–442.

[15] George, S. (2015). Hybrid PSO-MOBA for profit maximization in cloud computing. International Journal of Advanced Computer Science and Applications, 6(2), 159–163.