



# Improved Task Scheduling using Effective Particle Swarm Optimization in Cloud Computing Environment

Ankit Tomar, Bhaskar Pant, Vikas Tripathi, Priyank Pandey, Kamal Kant Verma

**Abstract:** A vibrant on demand service of today's era is cloud computing where one can utilize computer resources without indirect active management by user where one can use computing resources to achieve coherence in economic scale. Since cloud computing feel like Everything as a service so there should be highly scalable and reliable mechanisms to distribute the load evenly across the VMs evenly. Innumerable cloudlet mapping policies are presented in various research articles to achieve the high performance, better QOS and minimized task execution time but maximum are conventional approaches. No unconventional realistic scheduling algorithms is available which can schedule the tasks in heterogeneous manner. Since cloudlet scheduling is crucial metrics of cloud computing that has to be heightened by combining the different parameters. This paper tried to provide effectiveness and improvement in task scheduling using nature inspired Particle Swarm optimization (PSO) strategy. A powerful nature inspired load balancing mechanism is proposed in this paper which optimized makespan and throughput in environment of varying cloudlets and virtual machines results as compared to other conventional approaches. Proposed (EPSO) algorithm is with four scheduling policies namely FCFS, Round Robin (RR) and Shortest Job First (SJF) and get near twice good throughput percentage and minimized makespan in two different environments. Author used Cloud sim toolkit and some Open Source cloud packages to simulate the results of various scheduling components. Experimental results of various components are tested and simulated on java based CloudSim toolkit framework.

**Keywords:** Load Balancing, Particle Swarm, Cloud computing, CloudSim, Makespan, Task Scheduling.

## I. INTRODUCTION

From past few years cloud computing practiced lots of development particularly in the field of academics and industry. This has great potential of full utilization of elastic,

dynamic and virtualized resources. For virtualized resources a huge number of computing power is required in form of virtual machines, and hence thousand numbers of users would be engaged with the same, shown in figure 1. It is not easy to assign the tasks to cloud resources (virtual machines) manually since cloud is elastic environment so undoubtedly we required an efficient algorithm to map the tasks to the resources [1]. Since cloud environment is elastic and the scenario will change dynamically so first thing we must have highly scalable algorithm to schedule the tasks. The task mapping problem in cloud or grid computing belongs to a well-defined NP class problem which involves mapping. The main aim of task scheduler is to assign the task to the available resource according to the precedence with minimized make span. Make span is total time taken by the submitted task or we can say the overall execution time of the cloudlet that has to be submitted to and physical machine. Generally scheduling policies fall under two categories static and dynamic. In dynamic scheduling number of cloudlets (submitted jobs) varies with time which may cause overloading or under loading of VM. To spread the load equally across the physical machines and for taking advantage of full resource utilization and lessening the task completion time, essentially we need to design some effective load sharing algorithms. Scheduling algorithms are used to manage the load balancing problem that should be fair enough in order to distribute the load across the processors and for achieving the maximum throughput [11].

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Ankit Tomar**<sup>\*</sup>, CSE, Graphic Era Deemed to be University, Dehradun, India. Email: ankittomar.cse@gmail.com

**Bhaskar Pant**, CSE, Graphic Era Deemed to be University, Dehradun, India. Email: pantbhaskar2@gmail.com

**Vikas Tripathi**, CSE, Graphic Era Deemed to be University, Dehradun, India. Email: vikastripathi.be@gmail.com

**Priyank Pandey**, CSE, Graphic Era Deemed to be University, Dehradun, India. Email: priyankpandeyrc@gmail.com

**Kamal Kant Verma**, CSE, College of Engineering Roorkee, Roorkee India. Email: [kkv.verma@gmail.com](mailto:kkv.verma@gmail.com)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

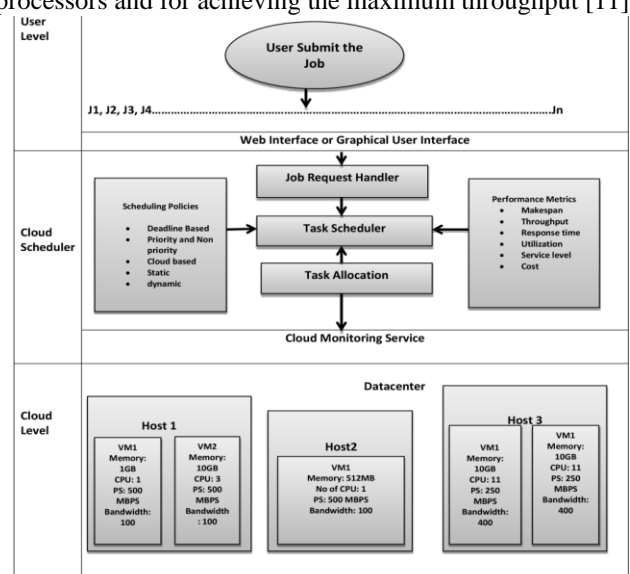


Figure 1. Essential Components required in task scheduling

A good scheduler always adapts dynamic environment that actually adapts the changing environment, and Particle swarm optimization is appropriate algorithm incorporate nature motivated process to schedule the tasks in dynamically.

Conventional algorithms (approaches) such as dynamic programming, divide and conquer, and branch & bound gives the universal optimum, but is over and over again time consuming so we cannot apply these of methods for resolving classic real-world problems. Hence researchers use optimal and dynamic schemes to solve the task assignment problems with minimal execution time and high throughput like ACO, GA and Lion optimization [2][7]. Two important units in cloud computing environment one is cloud providers other is cloud users, play an important role to establishing the smooth conduction of cloud services. If we look from the perspective of Cloud providers, it has enormous amount of computing resources in Data centers for renting out these resources to the users on pas go service by maximizing the greater revenue or profit by attaining high resource utilization. These resources are very dynamic in nature and at high demand for cloud users. To run cloud application in minimal expenses with varying load, cloud user lease the resources from cloud providers and utilizes the same in pay as go fashion.

II. BACKGROUND STUDY

We have three types of cloud services that are deployed to utilize on demand services anywhere and everywhere.

a. Software as a Service

SAAS service is provided and registered by service providers. Here users refer software related utilities without worrying fundamental techniques like how software’s services are accomplished. At SaaS level software is key resources where they get pooled among cloud users.

b. Platform as a Service

At PAAS level of cloud providers need to manage the platform regarding services underlying stages like hardware and software’s which allows user to organize and preserve the underlying applications without worrying about its availability.

c. Infrastructure as a Service

As name suggest this layer cloud providers prevails the essential building blocks of infrastructure. It afford the access the features of networking, IAAS also provide high level of control and flexibility to cloud

A. CloudSim Toolkit

CloudSim is most favorable simulator which is used to simulate the results and execute algorithms like own private cloud in the field of cloud computing [6].

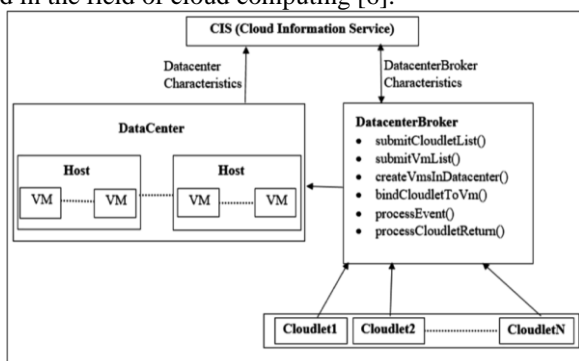


Figure 2. CloudSim Toolkit Framework

In earlier research work particularly from 2010 it is observed that majority of researchers and research papers adopted CloudSim toolkit to act out the cloud computing results since it provides real like cloud environment in a limited way[9]. Author has mentioned the essential cloud computing components which used in this paper are as follows [10].

B. CloudSim Entities

**Cloud Information Service:** CIS is a type of entity for registering the entities of data Centre along with discovery of cloud resources [13].

**Datacenter:** It is the class of homogeneous or heterogeneous cloud resources for hosting virtual resources. Its basic function is to deal with basic infrastructure (RAM, storage and bandwidth requirements) provided by service providers.

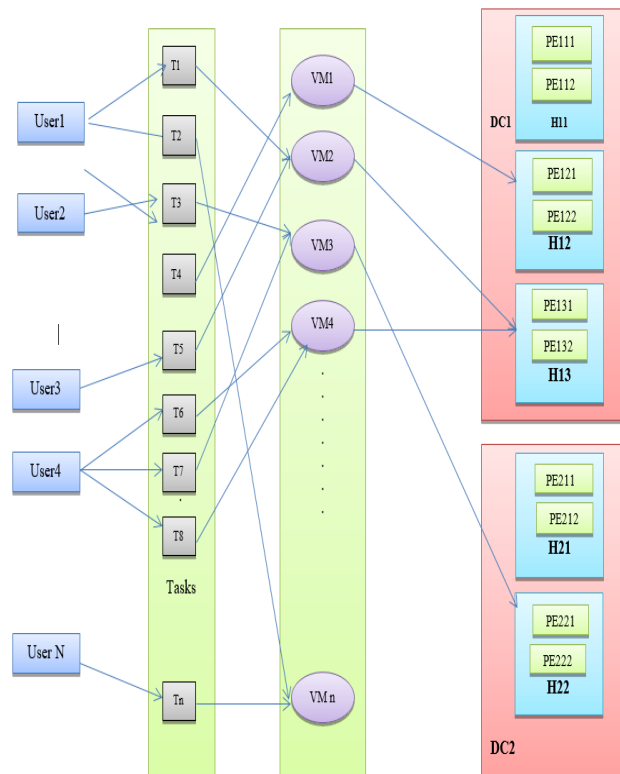


Figure 3. Pictorial view of scheduling the heterogeneous tasks

**Datacenter Broker:** Basically a class for users used for managing the VMs like creating, deletion and submission of VMs along with cloudlets submission.

**VM Scheduler:** Basically an abstract class which is responsible to break the VMs and hosts.

**Cloudlet Scheduler:** Type of abstract class for implementation of various policies time shared and space shared respectively for processing power of VMs among number of tasks (cloudlets).

**Cloudlet:** in cloud computing cloudlet is known submitted task on VMs.

**Host:** Again Host is like Cloudlets which may be homogeneous and heterogenous in nature with characteristics like cloudlet length, size of input file and output file.

**Virtual Machine:** is associated to hosts for sharing these hosts through other VMs.

All cloudlets are allocated to VMs by scheduler for accomplishment of execution. How all above components work together shown clearly in figure 2 and 3.

III. MOTIVATION AND RELATED WORK

Highly scalable resource arranging algorithms in cloud computing is on high demand so it is necessary to either work with new algorithm or to optimize the previous results. This research article demonstrates an enhanced particle swarm optimization (PSO) mechanism with adaptable transformation of parameters according to the advancement state-run assessment is presented. This variation helps to avoid precipitate convergence and explore the search space more efficiently [26]. Varying simulations are passed out to test proposed algorithm, test reveal that the algorithm can achieving significant optimization of makespan. Here we have presented the data of previous work done on particle swarm algorithm and we found that maximum articles covers either throughput or makespan at a time but in this paper author covered both scheduling metrics with good accuracy.

| CITE       | YEAR | PSO CATEGORY       | MAKESPAN | THROUGHPUT |  |
|------------|------|--------------------|----------|------------|--|
| [23]       | 2014 | Modified PSO       | ✓        |            |  |
| [17]       | 2019 |                    | ✓        |            |  |
| [21]       | 2018 | MultiObjective PSO | ✓        |            |  |
| [22]       | 2015 |                    | ✓        | ✓          |  |
| [18]       | 2018 | Standard PSO       | ✓        |            |  |
| [20]       | 2008 |                    | ✓        |            |  |
| [25]       | 2017 |                    | ✓        |            |  |
| This Paper |      |                    |          |            |  |
| [17]       | 2019 | Improved PSO       | ✓        | ✓          |  |
| [14]       | 2013 | Novel              | ✓        |            |  |
| [4]        | 2016 |                    | ✓        |            |  |
| [8]        | 2007 | Hybrid             | ✓        |            |  |
| [5]        | 2018 |                    | ACO+PSO  | ✓          |  |
| [19]       | 2016 |                    | GELS     | ✓          |  |
| [24]       | 2018 |                    | GA+PSO   | ✓          |  |
| [12]       | 2008 | Enhanced PSO       | ✓        | ✓          |  |

Table 1: old work done on PSO categories for Scheduling

P Visalakshi et al, 2009 proposed various versions of Particle swarm optimization PSO-fi, PSO-vi Elitism PSO (EPSO) and HPSO for task assignment problem (TAP) for heterogeneous tasks in non-pre-emptive environment, which is a NP hard problem [15].

Zhang et al, 2010 solved grid task scheduling algorithm using PSO, GA and CMPSO (cloud model PSO) for heterogeneous cloudlets, here author tried to balance the load very efficiently [3]. In this paper the correctness and performance of task scheduling algorithm is compared and showed that PSO balances the load very proficiently and observed that CMPSO gained lower running efficiency. The solution precision found better of CMPSO when compared with GA and PSO, However this paper is based on scheduling algorithms and discussed the correctness issues only, not any other load balancing parameter that is not satisfiable.

IV. PROBLEM FORMULATION

To maximize the utilization and profit, resource centric policy is beneficial and for cost and execution time application centric policy is best suited. An appropriate scheduling assures competent and fair allocation of resources, which are as follows:

**Makespan:** It is time taken by task from the beginning of process to end. It should be low for effective task scheduling policy [17], [21], [18], [20], [25], [11], [14], [8], [2], [13].

$$\text{Makespan} = \max(FT_j) \text{ where } j \in \text{VMs} \quad (1)$$

**Throughput:** For effective scheduling policy low throughput is required [11]. In cloud computing throughput relies number of cloudlets, which are run successfully with in required time [16], [13].

$$\text{Throughput} = \text{task}_j (\text{Execution Time}) \quad (2)$$

Particle swarm is a comprehensive nature call based optimization policy that works on concept of swarms in search of adaptation. PSO is very simple approach to solve NP complete problems; here variables adjust their values closer to closest member to any target at any particular moment [27]. In search of hidden food source circling a flock of swarm, one will chips food that is closest to it and other flocks swing nearby.

V. EXPERIMENTAL SETUP

Here in this paper author assumed submitted tasks are mutually heterogeneous (independent), i.e. there does not exists any precedence among the cloudlets hence no interfere of dependency of processors during execution. In current section we have provided some tabular form e.g. table 3 and 4, which is having numerical values of respective scheduling metrics against changing environment.

In this paper author used java based platform using ide eclipse, basically it platform to run simulation tool CloudSim toolkit version 3.03, with some jar files. Results are computed with some number of datacenters with some hosts, where every host having certain characteristics. Every entity and its attributes are defined in lucid manner which is shown in below table.

Table 2: Experimental Setting and parameters

| Entities        | Parameters           | Values              |
|-----------------|----------------------|---------------------|
| Cloudlet        | Length of Cloudlet   | [1000-2000] (MI)    |
|                 | Number of Tasks      | [50-1000]           |
|                 | Tasks MIPS           | [500-10000]         |
| Virtual Machine | VM RAM size          | [512-1024] MB       |
|                 | VM Policy Used       | Time & Space Shared |
|                 | Number of VMs        | 20                  |
|                 | Processing Elements  | [1-5]               |
|                 | VM MIPS              | [100-1000]          |
|                 | Bandwidth            | 1000                |
|                 | Memory Capacity      | [256-2048]MB        |
|                 | VMM                  | XEN, Linux          |
| Hosts           | Number of Hosts      | 2                   |
|                 | RAM size             | 2018MB              |
|                 | Memory               | 1000000             |
|                 | Policy               | Time & Space Shared |
|                 | Bandwidth            | 10000               |
| DC              | Number of DC         | 3                   |
| Iteration       | Number of iterations | [100-1200]          |

VI. RESULT ANALYSIS AND DISCUSSION

A. Evaluation of EPSO with different parameters

We implemented proposed EPSO algorithm by simulation author tried to investigate its weakness and strengths.



We have considered crucial PSO parameters (s, w, L1, L2, number of iterations) which impacts openly or indirectly its results, where L1 and L2 are swarm learning factors, w is inertia coefficient and s is particle index. Through simulation results we computed several results on each parameter with the help of CloudSim.

In figure 4 range of submitted task is taken from 100 to 1000, against range of submitted cloudlets we observed that minimum execution time of tasks is increasing while population size is kept 25, which means to improve the execution time (makespan) in similar scenario need to change the population size of proposed PSO scheme.

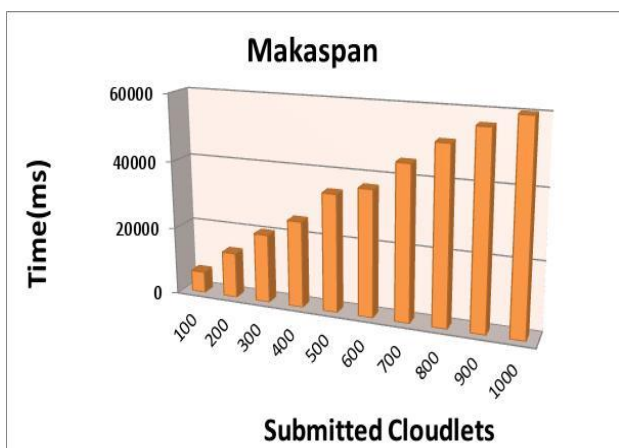


Figure 4. Makespan calculated against submitted tasks

Table 3: Makespan, best fitness value for increasing population size

| Population Size | Alpha | Makspan     | Fitness Value | Tasks |
|-----------------|-------|-------------|---------------|-------|
| 25              | 0.1   | 4894.139274 | 8570.44197    | 100   |
| 25              | 0.2   | 6004.184622 | 12945.42795   | 100   |
| 25              | 0.4   | 8588.038865 | 19732.7089    | 100   |
| 25              | 0.6   | 10776.44788 | 26489.60979   | 100   |
| 25              | 0.8   | 15365.18247 | 32983.85513   | 100   |
| 25              | 1     | 18651.84998 | 39882.2885    | 100   |

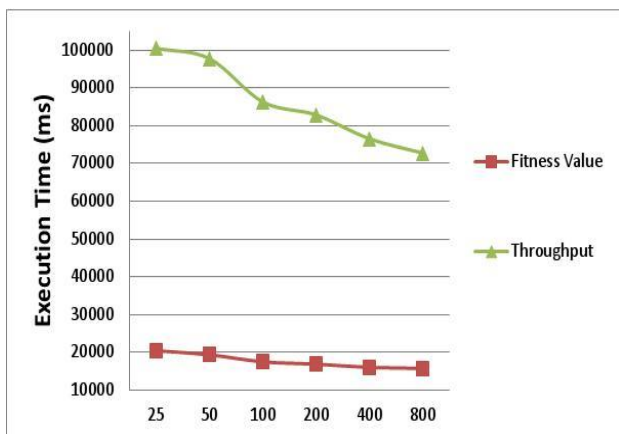


Figure 5: Fitness value and throughput against no of iterations

Table 4: Fitness value and throughput for increasing Simulations

| No. of Iterations | Fitness Value | Throughput | Cloudlets |
|-------------------|---------------|------------|-----------|
| 25                | 20425.94605   | 100531.592 | 100       |
| 50                | 19338.51155   | 97800.278  | 100       |
| 100               | 17475.68637   | 86339.352  | 100       |
| 200               | 16857.2155    | 82882.784  | 100       |
| 400               | 15997.3839    | 76584.202  | 100       |
| 800               | 15735.32811   | 72773.25   | 100       |

Exploration and Exploitation Factors: in EPSO equations 3 and 4,  $X_i$  is basically position vector which actually fix the position of particles, velocity vector  $VV_{id}$  fix the direction and intensity of particle movement.

$$VV(t+1) = w * VV(t) + L1 * r1 * (Xpbest(t) - X(t)) + L2 * r2 * (Xgbest(t) - X(t)) \quad (3)$$

$$X(t+1) = X(t) - VV(t+1) \quad (4)$$

**B. Implementation of EPSO with FCFS, RR and SJF**

**i. Comparison of Throughput**

Throughput is computed for variable VMs and cloudlets, equation 2 illustrates how it is computed in dynamic cloud environment. In both cases line of EPSO falls comparatively down side which is depicted in Figure 6, 7). Since under EPSO lower throughput measured and low throughput is desired for effective scheduling in cloud computing.

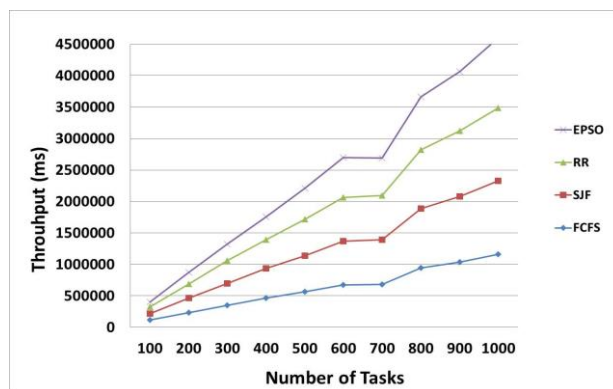


Figure 6. Throughput Calculated for varying no of tasks

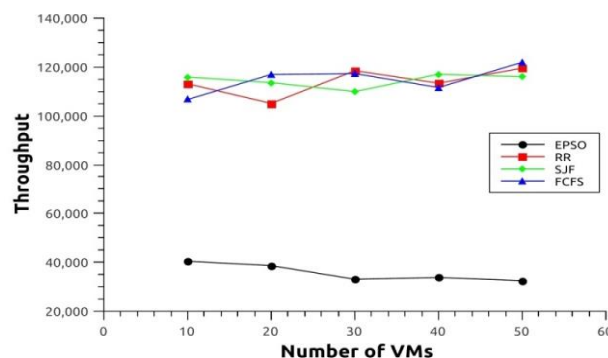


Figure 7. Throughput Calculated for varying no of VMs

**ii. Comparison of Makespan**

Makespan for all algorithms is calculated for increasing VMs and number of cloudlets, equation 1 illustrates better.



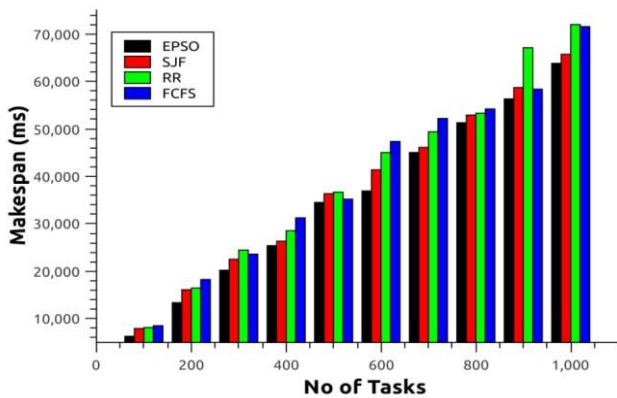


Figure 8. Makespan Calculated for varying no of tasks

Under particle swarm algorithm makespan is recorded lesser in each and every situation which is shown over bar graphs in figure 8 and 9. From results, EPSO is supposed to be a healthy scheduling policy.

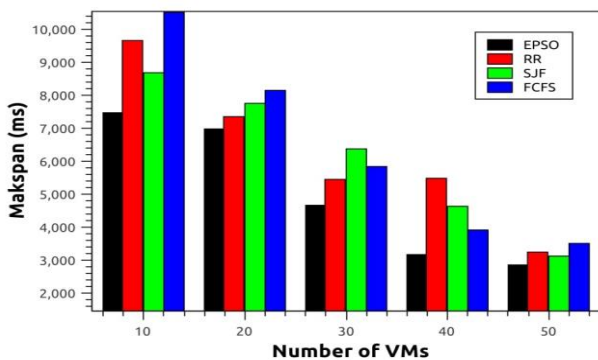


Figure 9. Calculated Makespan for varying VMs

## VII. CONCLUSION

Over recent years cloud computing model gained popularity by developing high performance load balancing models by maximizing the throughput, therefore users now could able to use cloud services paying with ease comparatively lower computing charges. Since energy consumption is directly proportional to hardware utilization therefore we are still in search of better scheduling policy which doesn't have balance between makespan and throughput to maximize the profit for server and client. The crucial objective of this paper fill this trade-off gap through dynamic mapping of cloudlets to appropriate virtualized resources. Various static scheduling policies like FCFS, SJF, RR and effective PSO are used here. Paper demonstrates optimal values of makespan and throughput under EPSO. The comparative study of all variants of PSO has also been discussed where we found that none PSO based research papers till now worked on two or more than two scheduling parameters but in this paper we tried to cover the maximum crucial scheduling terms which enhances the novelty of this paper

## REFERENCES

1. Qiyi, H., Tinglei, H., "An Optimistic Job Scheduling Strategy based on QOS for Cloud Computing" in 2010 IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), DOI: 10.1109/ICISS.2010.5655492, pp.673-675, 2010.
2. Tawfeek, Medhat A., "Cloud task scheduling based on ant colony optimization." 2013 8th international conference on computer engineering & systems (ICCES). IEEE, 2013.

3. Zhong, Shao Bo, and Zhong Shi He. "The scheduling algorithm of grid task based on PSO and cloud model." Key Engineering Materials. Vol. 439. Trans Tech Publications, 2010.
4. Milani, Alireza Sadeghi, and Nima Jafari Navimipour. "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends." Journal of Network and Computer Applications 71 (2016): 86-98.
5. Tomar, Ankit, et al. "Dynamic Task Migration Mechanisms in Cloud Environment: Literature Review and Future Trends." Recent Trends in Science, Technology, Management and Social Development (2018): 55.
6. Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41.1 (2011): 23-50.
7. Verma, Kamal Kant, Pradeep Kumar, and Ankit Tomar. "Analysis of moving object detection and tracking in video surveillance system." 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2015.
8. Sidhu, et al. "A load-rebalance PSO heuristic for task matching in heterogeneous computing systems." 2013 IEEE Symposium on Swarm Intelligence (SIS). IEEE, 2013.
9. Kumar AM Senthil, M. Venkatesan. "Task scheduling in a cloud computing environment using HGPSO algorithm." Cluster Computing (2018): 1-7.
10. Tareghian, Shahab, and Zarintaj BORNAEE. "A new approach for scheduling jobs in cloud computing environment." Fen Bilimleri Dergisi (CFD) 36.3 (2015).
11. Madni, et al. "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review." Cluster Computing 20.3 (2017): 2489-2533.
12. Yin, Peng-Yeng, et al. "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems." Computer Standards & Interfaces 28.4 (2006): 441-450.
13. Ijaz, Samia, et al. "Efficient scheduling strategy for task graphs in heterogeneous computing environment." International Arab Journal of Information Technology. 10.5 (2013): 486-492.
14. Pooranian, Z., et al. "Hybrid pso for independent task scheduling in grid computing to decrease makespan." Proc. of International Conference on Future Information Technology, IPCSIT'11. Vol. 13. 2011
15. Visalakshi, P., and S. N. Sivanandam. "Dynamic task scheduling with load balancing using hybrid particle swarm optimization." Int. J. Open Problems Compt. Math 2.3 (2009): 475-488.
16. Madni, Syed Hamid Hussain, et al. "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment." PloS one 12.5 (2017): e0176321.
17. Wang, Shangguang, et al. "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers." 2013 International Conference on Parallel and Distributed Systems. IEEE, 2013.
18. Zhang, Guoxiang, and Xingquan Zuo. "Deadline constrained task scheduling based on standard-PSO in a hybrid cloud." International Conference in Swarm Intelligence. Springer, Berlin, Heidelberg, 2013.
19. Madni, Syed Hamid Hussain, et al. "An appraisal of meta-heuristic resource allocation techniques for IaaS cloud." Indian Journal of Science and Technology 9.4 (2016): 1-14.
20. Guo, et al. "Task scheduling optimization in cloud computing based on heuristic algorithm." Journal of networks 7.3 (2012): 547
21. Ramezani, et al. "Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization." International Conference on Service-oriented computing. Springer, Berlin, Heidelberg, 2013.
22. Natesan, Gopalakrishnan, and Arun Chokkalingam. "Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm." ICT Express (2018)
23. Jana, et al. "A Task Scheduling Technique Based on Particle Swarm Optimization Algorithm in Cloud Environment." Soft Computing: Theories and Applications. Springer, Singapore, 2019. 525-536.
24. Jena, R. K. "Multi objective task scheduling in cloud environment using nested PSO framework." Procedia Computer Science 57 (2015): 1219-1227.
25. Zuo, et al. "Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud." IEEE Transactions on Automation Science and Engineering 11.2 (2014): 564-573.

26. Sewaiwar, Purva, and Kamal Kant Verma. "Comparative study of various decision tree classification algorithm using WEKA." International Journal of Emerging Research in Management & Technology 4 (2015): 2278-9359.
27. Verma, Kamal Kant, Brij Mohan Singh, and Amit Dixit. "A review of supervised and unsupervised machine learning techniques for suspicious behavior recognition in intelligent surveillance system." International Journal of Information Technology (2019): 1-14.

## AUTHORS PROFILE



**Mr. Ankit Tomar** has done B Tech Computer Science and Engineering from G. L. Bajaj Institute of Technology, UPTU, M-Tech in from Jamia Hamdard, New Delhi and pursuing PhD from Graphic Era Deemed University, Dehradun. He is actively involved in research

related to Cloud Computing and Machine Learning. He has published many papers in reputed international conferences and journals. Currently he is working as an assistant professor in Graphic era deemed to be university Dehradun, India.



**Dr. Vikas Tripathi** has done BE in information technology from Technocrats institute of technology, Bhopal, M. Tech in Software engineering from Indian institute of information technology Gwalior and PhD from Uttarakhand technical university, Dehradun. He is actively involved in research related to Software

engineering, Computer Vision, Machine learning and Video Analytics. He has published many papers in reputed international conferences and journals. Currently he is working as an associate professor in Graphic era deemed to be university Dehradun, India.



**Dr. Bhaskar Pant** Currently working as Dean Research & Development and Associate Professor in Department of Computer Science and Engineering. He is Ph.D. in Machine Learning and Bioinformatics from MANIT, Bhopal. Has more than 15 years of experience in Research and Academics. He has till now guided as Supervisor 3 Ph.D. candidates (Awarded).and 5 candidates are in

advance state of work. He has also guided 28 MTech. Students for dissertation. He has also supervised 2 foreign students for internship. Dr.Bhasker Pant has more than 70 research publication in National and international Journals. He has also chaired a session in Robust Classification & Predictive Modelling for classification held at Huangshi, China.



**Priyank Pandey** has done B Tech in Computer Science from Amity University, Lucknow, M. Tech in Computer Science from Guru Gobind Singh Indraprastha University Delhi. He is actively involved in research related to Natural Language Processing, Machine Learning,

software engineering, Computer Vision and Video Analytics. He has published many papers in reputed international conferences and journals. Currently he is working as an Assistant Professor in Graphic era university Dehradun, India.



**Third Kamal Kant Verma** is research scholar in Uttarakhand Technical University Dehradun. He is currently working as an Assistant Professor in Department of CSE, COER Roorkee Uttarakhand India.

He has publish many years of Teaching and research experience. His research area is Computer Vision, Human Activity Recognition, and Deep Learning.