QoS Optimization in Cloud Computing Networks

Abhikriti Narwal[a], Sunita Dhingra[b]

[a,b] Department of Computer Science and Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak
[a] abhikritiin@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract: As an increasing era of virtualization, resources are considered as services. To achieve the efficient utilization of the resources,a proper Scheduling and Load balancing technique is required on the Cloud Network. Quality of service parameters plays a vital role in achieving this good goal of utilization. Different load balancing and scheduling strategies of cloud computing studies held to survey QoS optimization. This paper aims at QoS efficiency to develop scheduling and load balancing algorithms to achieve optimization in the cloud networks. This paper briefly talks about the computing model concepts and QoS optimization study executed in various algorithms.In this paper, the performance optimization comparison of improvised algorithms CBSA_LB, EMOSA, and EMOSA_LB with base algorithms MOSA and CBSA has been carried out. The results show that the improvised version has significantly improved the optimization percentage of all performance parameters.

Keywords: Cloud Computing, QoS Optimization, Quality of services, task scheduling

1. Introduction

Cloud computing in this modern era is a field of High-Performance Computing Problems (HPC) [1]. The improper task scheduling or application in the data centre results in the underutilization of cloud resources—the underutilization of any application in the cloud results in a violation of service level agreement. Cloud computing supplies efficient resources with self-managed virtual infrastructure on-demand requests [2]. The resource allocation model based on QoS uses the fitness function to increase the gain and improve energy consumption, task rejection ratio, and execution time. QoS parameters, namely availability, reliability, degree of imbalance, and SLA violation, are required for exemplary cloud service [3].

In the area of Quality-Of-Service (QoS) management, the cloud creates a new issue. The QoS area portrays fields of cloud-like reliability, performance, platform, and availability offered by a particular application and hosts. With the advancements of cloud computing, QoS properties gain continuous attention. A notable mix-up of QoS prediction, assurance, and analysis occurs in Heterogeneity and resource isolation mechanisms of cloud platforms [4].

This paper investigates various heuristics in cloud computing to analyze the various QoS like makespan time, throughput, processing cost, processing time, execution time, and response time. Performance analysis of the various heuristics is done to determine the optimization achieved by the developed heuristics over the existing heuristics. The analysis of the developed algorithms CBSA-LB, EMOSA-LB, and EMOSA is done with the existing algorithms in the literature that are MOSA and CBSA. Different scenarios of cloudlets and virtual machines are taken forthe readings of the developed work. For each algorithm, five different schedules are taken with varying tasks and virtual machines. From the comparison of these five task scheduling and load balancing algorithms, it is observed that the developed algorithms EMOSA, EMOSA_LB, and CBSA_LB performs better than the base algorithms MOSA, CBSA with minimum makespan time, processing time, execution time, average response time, processing cost and increased throughput of the cloud computing system.

The paper is arranged to analyze Section 2 analyses the related work, and Section 3 talks about various heuristics of system models utilized here. Then Section 4 narrates workload characteristics, experimental setup, discussion relates to gain performance and performance evaluation. Finally, Section 5 completes with the conclusion and targets future directions of this study.

2. Related Work

QoS optimization is trending in virtual networks these days to achieve the maximum utilization of resources. The computing models with high Quality of service (QoS) were proposed with increased demands from service users. The work in computing in various areas, namely service level agreement, server optimization, scientific workflow execution, resource management, allocation, and load balancing,has been done.

Lakra et al.[2] proposed a multiobjective task scheduling algorithm to map the tasks to VMS. The data centre was improved, and the cost was reduced when service level agreement was followed in the application of cloud SaaS with the obtained environment.The author has calculated the values of the turnaround time of three different scheduling algorithms, namely First come, First serve, Priority scheduling, and Multiobjective scheduling. The comparative results showed that the proposed algorithm had performed better than the other two algorithms with minimum execution time and increased throughput of the cloud computing system.

Honeymol et al.[5] examined a new approach with Ant- colony optimization where a graph-based approach is used, maximizing the QoS and reducing the cost. The previous approach concentrated on profit increase but not on the QoSsomewhat satisfied customer service, which is mainstream.

Abhikriti et al. [6] developed a Task Scheduling Algorithm using Multiobjective functions for a cloud computing environment. The tasks are arranged according to task length, input file size, and output file size and then assigned to the particular virtual machine based on their MIPS and Granularity Size. The three parameters, namely total processing time, total processing cost, and average waiting time, were optimized using the developed algorithm. It also succeeds in showing the improved performance values over the previously defined algorithms.

Sirisha Potluri et al.[7] examined the parameters used for scheduling and the QoS-based task scheduling. A wide variety of task scheduling algorithms have been discussed that addressed the security problems in cloud computing when used with Quality of service. They have compared the results and limitations of various discussed algorithms. For the improvisation of QoS based on the Task scheduling algorithm, various factors, namely task arrival time, execution on resource, and cost for communication was considered for different tasks.

Abhikriti et al. [8] intimated the algorithm for scheduling that gave a good performance for proper allocation of tasks using various factors. They proposed a multiobjective scheduling algorithm to improve the performance of cloud networks. They have considered the different parameters while scheduling their total processing cost, total processing time, and average waiting time. They have considered different scheduling conditions to run the algorithm under different scenarios to obtain better results than the previously defined algorithms.

Antony Thomas et al. [9]discussed such algorithms that favour user satisfaction and resource availability. The algorithm put forward considers both task length for scheduling tasks on the cloud network and user priority.It shows improvement in the utilization of resources.

K R Babu et al. [10] put a honey bee colony-based algorithm for an efficient load balancing technique. The hunting behaviour of honey bees was examined to balance the Load among virtual machines in the cloud network. They have significantly improved the Quality of service(QoS) by reducing the minimum response time and fewer task migrations between the virtual machines. In this work, the tasks are removed from the overloaded virtual machines and moved to the underloaded ones. They have considered overloaded Virtual machines as honey bees and under-loaded virtual machines as food sources.

Abhikriti et al. [11]performed a comparative analysis between the two scheduling algorithms. The comparison between Credit Based Scheduling Algorithm (CBSA) and the Multiobjective scheduling algorithm (EMOSA) was considered. The used algorithms were tested in different scenarios and results shown that such a study can help the designer define good scheduling algorithms that can help improve the various Quality of Service parameters like time and cost over the cloud network.

To balance the Load among the virtual machines while scheduling the tasks, Abhikriti et al.[12] put forward a CBSA_LB algorithm. It balanced the Load equally and solved the problem of overloaded and underloaded virtual machines. It assessed the output on six parameters: processing cost, processing time, response time, makespan time, Execution, and Throughput Time.

Zhiguo Qu et al. [13] made a study on QoS optimization and energy saving in cloud computing, edge computing, fog computing, and IoT environment. The study summarized the main trouble with these techniques and analyzed the respective key to solving. The work proposed energy-saving techniques and QoS optimization in those models, which helped readers perceive different computing models.

Altaf Hussain et al. [14] put forth an algorithm based on the computation capabilities of VM.s. It balanced the distribution of workload termed as Resource Aware Load Balancing Algorithm (RALBA). The RALBA working module was patriated into scheduling tasks on computing capabilities of V.M.s as one partition and V.M. with the earliest finish time for job mapping as the second partition. The algorithm used two schedulers to schedule the work within V.M.s, namely SPILL and FILL. The traditional heuristics that heed makespan, throughput, and resource utilization were considerably improved when viewed from the RALBA framework.

Abhikriti et al. [15] intimated an Enhanced Multiobjective Load balancing Scheduling Algorithm (EMOSA_LB) where Honey Bee colony optimization was used load balancing of tasks. More parameters were used to sort and upgrade the performance over the cloud network termed time and cost. They have also used the EMOSA algorithm to sort the tasks developed by the same author. The obtained results were analyzed and compared with various existing techniques like MOSA, EMOSAand this proposed technique outperforms all the existing techniques in terms of various parameters like average waiting time by 2.934%, processing cost by 17.6%, and processing time by 20.5%.

3. Methodology

The task scheduling is performed on the cloud network for optimization and to produce an optimized system simultaneously.To achieve this goal, various scheduling algorithms are implemented and improved accordingly the problems occurred. In this paper, the study of scheduling algorithms has begun with the Multiobjective Scheduling Algorithm[2]. The shortcomings that have occurred in the MOSA scheduling algorithm were replaced by a new scheduling algorithm which was named EMOSA. Later on, features of load balancing are clubbed with EMOSA and named EMOSA_LB.EMOSA faced the problem when two tasks arrived with the same priority; to resolve this problem of priority and obtain better performance characteristics, the study of credit-based scheduling algorithm is made and termed this algorithm as CBSA. CBSA lacked the concept of load balancing, which is further defined with the new algorithm credit-based scheduling algorithm with load balancing (CBSA_LB). All algorithms taken for analysis of a study of QoS are described as follows:

Multiobjective Scheduling Algorithm (MOSA)

Multiobjective Task Scheduling Algorithm considers two parameters for its implementation that are cloudlet length and QoS. Based on these two parameters, tasks are sorted, and a cloudlet list is created. This cloudlet list is then submitted for further processing on Virtual machines arranged based on MIPS [2].

Enhanced Multiobjective Scheduling Algorithm (EMOSA)

In the Enhanced Multi-Objective Scheduling Algorithm, the parameters for implementing tasks are extended to cloudlet length, input file size, and output file size. Based on these parameters, different tasks are arranged, and a cloudlet list is created. The sorted cloudlist is then submitted for further processing on Virtual Machines which are arranged based on MIPS and Granularity Size[8].The algorithm of EMOSA, which is based on MOSA [2] form literature, is as given below:

- Initializing the cloudsim library.
- Create datacenters, hosts, and processing elements list, Virtual machines.Tasks are created by calling V.M. and Cloudlet constructor.
- Sort the created V.M. list using MIPS and grouping factor
- Apply sorting based on cloudlet dominates other cloudletsbased on cloudet_length, fileinputsize, fileoutputsize and then make the decision accordingly and add to the sorted list.
- Submit sorted List and sorted VMList to Broker and start the simulation
- Mapping of cloudlet to the VM. is done
- Analyze the performance parameters for each cloudlet processing on V.M.

Enhanced Multiobjective Scheduling Algorithm with Load Balancing(EMOSA_LB)

In Enhanced Multiobjective Scheduling Algorithm with Load Balancing, previously defined EMOSA[8] algorithm is combined with honey bee load balancing technique. Based on the parameters of EMOSA [8],tasks are arranged, and a cloudlet list is prepared. The sorted Cloudlet list is submitted to process these sorted cloudlets on virtual machines based on the load balancing mechanism of the honey bee algorithm in which the Load of the machine is verified by determining its capacity (No. of Tasks).

The honey bee's load balancing mechanism deals with redistributing a total load of a distributed system into individual nodes to ensure that no node is overloaded and no nodes are under-loaded or idle. Load balancing is a good backup in case the virtual machines failovers in the cloud network.[8,10]

Figure 1. Depicts the sequence of work carried out to study different scheduling algorithms with their advantages and disadvantages. This work sequence is later used to find out the performance percentage improvement analysis among these different algorithms.

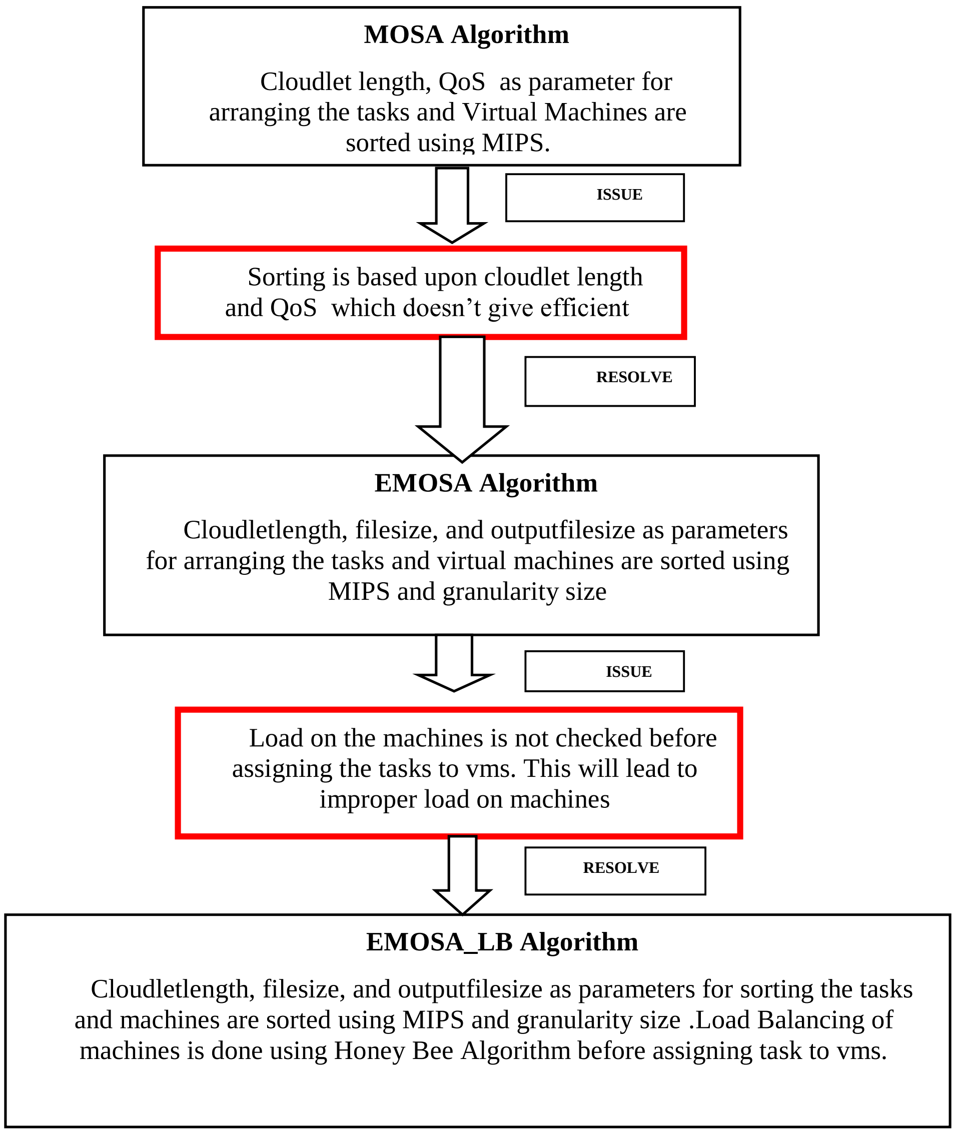

MOSA Algorithm
Cloudlet length, QoS as parameter for arranging the tasks and Virtual Machines are sorted using MIPS.
ISSUE
Sorting is based upon cloudlet length and QoS which doesn't give efficient
RESOLVE
EMOSA Algorithm
Cloudletlength, filesize, and outputfilesize as parameters for arranging the tasks and virtual machines are sorted using MIPS and granularity size
ISSUE
Load on the machines is not checked before assigning the tasks to vms. This will lead to improper load on machines
RESOLVE
EMOSA_LB Algorithm
Cloudletlength, filesize, and outputfilesize as parameters for sorting the tasks and machines are sorted using MIPS and granularity size .Load Balancing of machines is done using Honey Bee Algorithm before assigning task to vms.

Figure1: Sequential flow of Multi-Objective Scheduling algorithms with problems and improvisation

Credit-based Scheduling Algorithm (CBSA)

CBSA scheduling is based on credits assigned to a task to overcome the issues involved in EMOSA[8] and MOSA[2] ,as after sorting of tasks, there are some cases left in which some tasks have the same priority, in that case, the tasks arrived first will get the virtual machine, but this will impact the performance parameters. To solve this issue of having the same priority, additional credits are combined to sort the tasks to have the same priority. Credits taken are cloudlet length, deadline, priority, and cost. Based on these credits, tasks are sorted and then submitted to process the tasks on the virtual machines further and analyze the parameters. [12]

Credit-based Scheduling Algorithm with Load Balancing(CBSA-LB)

CBSA-LB scheduling is based on sorting the tasks using different credits and then gives those tasks to virtual machines, which are sorted based on the load balancing mechanism of the honey bee load balancing technique. The load balancing mechanism of the honey bee ensures that no V.M.s are overloaded, where some machines are underloaded or doing very little work by transferring extra Load to these virtual machines. Load Balancing mechanisms try to speed up the execution time of applications and system stability. CBSA-LB algorithm is based on CBSA from the literature [9].

Credit-Based Scheduling Algorithm :

Procedure 1: Credit based on Length of task

- For all submitted tasks in the set; Ti
- Task length difference (TLD) = absolute value (average Length – Length of particular task)
- If $\boldsymbol{TLD_i} \leq$ value_1
- then credit =5
- else if value_1 $< \boldsymbol{TLD_i} \leq$ value_2
- then credit =4
- else if value_2 $< \boldsymbol{TLD_i} \leq$ value_3
- then credit =3
- else if value_3 $< \boldsymbol{TLD_i} \leq$ value_4
- then credit =2
- else value_4 $> \boldsymbol{TLD_i}$
- then credit =1
- End For
- where value_1= high_len / 5;
- value_2= high_len / 4;
- value_3=value2+value1;
- value_4=value3+value2;

Procedure 2: Priority credits assigning to task

- For each submitted tasks in set; Ti
- Search for highest priority task
- Choose the divisible factor for priority
  $$\boldsymbol{credit\_Priority_i = TaskPriority_i / divisiblefactor_i}$$
- End For

Procedure 3: Deadline of the task

- For each submitted tasks in the set; Ti
- Search for MAXMIPS of the V.M. from the virtual machine list
  $$\boldsymbol{Deadline\_Task_i = Credit\_Length_i * Credit\_Priority_i / MIPS_{MAx}}$$
- End For

Procedure 4: Cost of the task

- For each submitted tasks in set; Ti

$$\boldsymbol{Cost\_Task_i = DC_{characteristics}.Costpermemory * Vm_{Ram} + DC_{characteristics}.Costperstorage * Vm_{size}}$$

- End For

Procedure 5: Honey bee optimized algorithm

- For all V.M. in set; $\boldsymbol{Vm_i}$
- While termination condition is false perform
- Find the Load on the virtual machine using

$$\boldsymbol{Load_i = \frac{N * Cloudlet\_length}{VM\_MIPS}}$$

Where N is number of tasks assigned to a VM, *Cloudlet_length* is the Length of single tasks and $\boldsymbol{VM\_MIPS}$ is the MIPS rate of that V.M.

- The capacity of a particular V.M. could be computed using

$$\boldsymbol{Capacity_i = PE_{Num} * PE_{MIPS} + Vm_{BW}}$$

Where $\boldsymbol{Capacity_i}$= Processing capacity of $\boldsymbol{Vm_i}$

$\boldsymbol{PE_{Num}}$ = Number of Processor of V.M.

$\boldsymbol{PE_{MIPS}}$=Million Instructions per second of processor of V.M.

$\boldsymbol{Vm_{BW}}$=Bandwidth of Virtual Machine

- By using Load and capacity of a V.M., processing time can be computed by equation

$$ProcessingTime_i = \frac{Load}{Capacity}$$

- Finally, by utilizing the equation below Standard deviation (S.D.) of Load can be computed as,

$$SD = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(X_i - \overline{X})^2}$$

Where X_i is Processing Time and $\overline{X}$ is the average Processing Time of the virtual machine.

- Based on the above calculation mark VM's as under-loaded or over-loaded.
- In this step, underloaded and overloaded VMs will be calculated.
- Arrange the under-loaded and over-loaded VM sets based on Load.
- Arrange the tasks in over-loaded VM.s based on priority.
- For every task in all over-loaded VM search an appropriate under-loaded VM.
- Revise the under-loaded and over-loaded VM sets and repeat step 2.
- End while

Figure 2. Depicts the sequence of work carried out to study different scheduling algorithms with their advantages and disadvantages. This work sequence is later used to find out the performance percentage improvement analysis among these different algorithms.

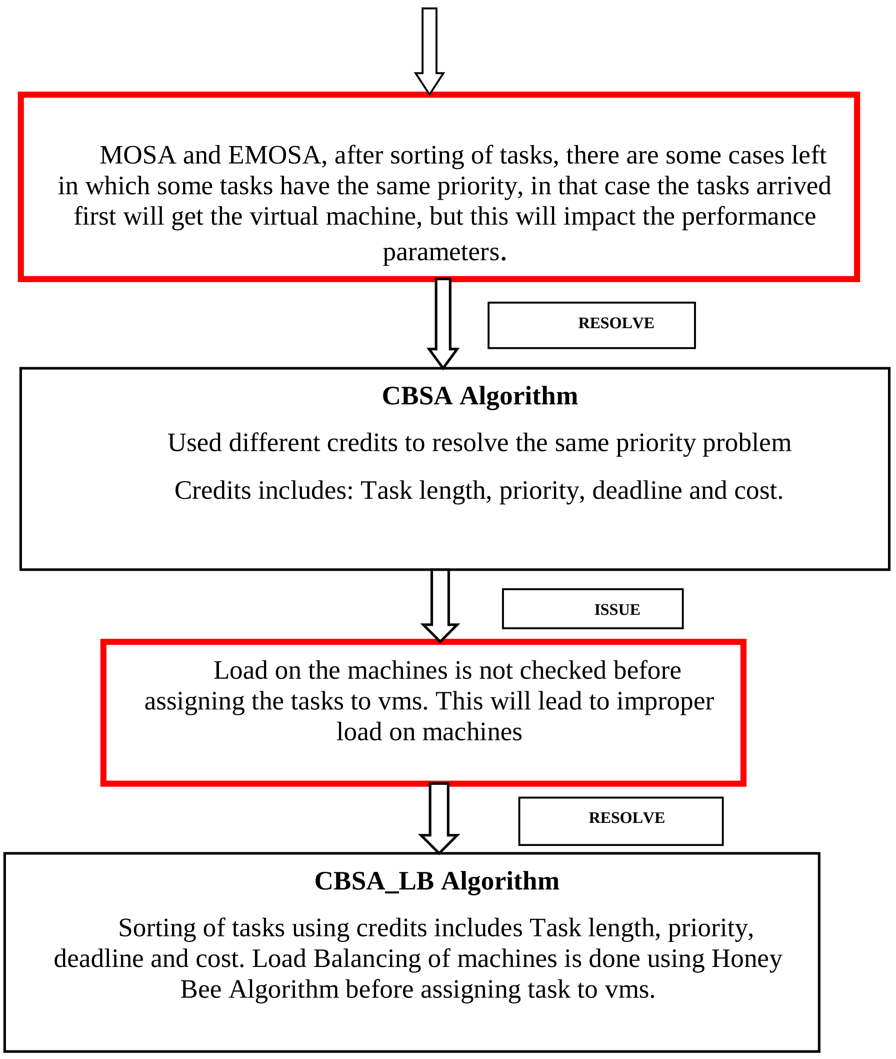

MOSA and EMOSA, after sorting of tasks, there are some cases left in which some tasks have the same priority, in that case the tasks arrived first will get the virtual machine, but this will impact the performance parameters.
RESOLVE
CBSA Algorithm
Used different credits to resolve the same priority problem
Credits includes: Task length, priority, deadline and cost.
ISSUE
Load on the machines is not checked before assigning the tasks to vms. This will lead to improper load on machines
RESOLVE
CBSA_LB Algorithm
Sorting of tasks using credits includes Task length, priority, deadline and cost. Load Balancing of machines is done using Honey Bee Algorithm before assigning task to vms.

Figure2: Sequential flow of Credit based Scheduling algorithms with problems and improvisation.

4. Results and Discussion

The developed algorithms have been implemented in Cloudsim 3.0 framework using Netbeans 8.0 as an IDE to run the Cloudsim simulator. The developed techniques have been executed on various datasets comprising three virtual machines viz 40,60 and 80 machines with different sets of tasks ranging from 100 to 2000. Performance parameters are calculated for each scenario with parameters namely, Makespan Time (M.S.), Total Execution Time (TET), Average Response Time (ART), Throughput (T.P.), Total Processing Time (TPT), and Total Processing Cost (TPC). In this paper, the improvement percentage of the developed multiobjective algorithms and credit-based algorithms are compared concerning their base algorithms, and improvement percentage is calculated and analyzed in terms of comparison graphs. The equation for calculating the improvement percentage is mentioned below:

$$Improvement\ \%age = \left(\frac{Base\ Algorithm - Proposed\ Algorithm}{Base\ Algorithm}\right) * 100$$

Table 1: Improvement %agecomparison of Improvised Multiobjective Algorithms concerning MOSA

Algorithms	MST	TET	TP	ART	TPT	TPC
EMOSA	38.08	14	60.45	28.52	14.02	49.72
EMOSA-LB	49.41	16.21	96.6	31.44	78.4	92.8

MOSA

Table 1. show the comparison results of Minimum Spanning Time (MST),Total Execution Time (TET),Throughput(T.P.),Average Response Time (ART),Total Processing time(TPT), and Total Processing Cost(TPC) of the algorithms EMOSA [8] and EMOSA_LB[15] with the base algorithm MOSA[2].

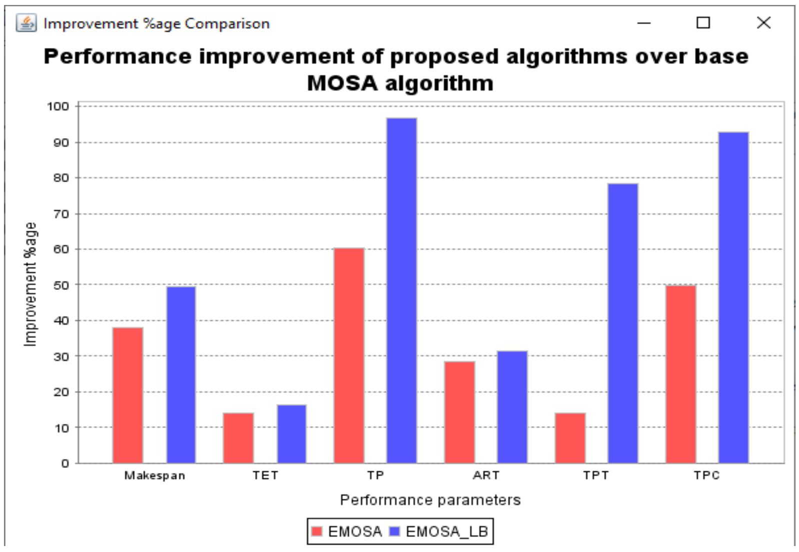
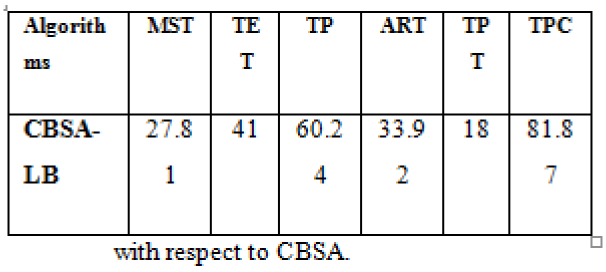


Figure 3: Improvement % age comparison of improvised Multiobjective algorithms with respect to MOSA

Table 2: Improvement %age comparison of improvised Credit- based scheduling algorithm with respect to CBSA.



Algorithms	MST	TET	TP	ART	TPT	TPC
CBSA-LB	27.81	41	60.24	33.92	18	81.87

with respect to CBSA.

Table 2. show the comparison results of Minimum Spanning Time (MST),Total Execution Time (TET),Throughput(T.P.),Average Response Time (ART),Total Processing time(TPT), and Total

Processing Cost(TPC) of the algorithms CBSA_LB[12] with the base algorithm CBSA[16].

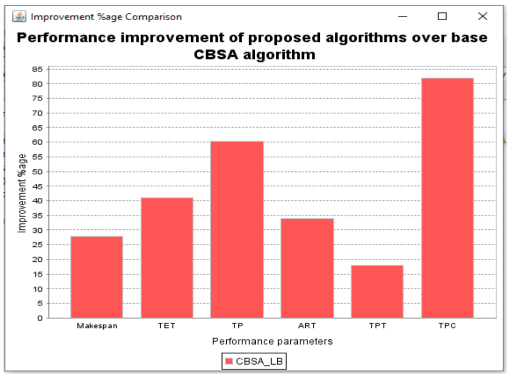


Figure 4: Improvement % age comparison of improvised algorithm CBSA_LB concerning CBSA

Figures 3and 4 above show the % age improvement of the developed algorithms concerning the base algorithm. The makespan time improvement in developed algorithms EMOSA_LB and EMOSA is 49.41% and 38.08%, respectively, to the base technique MOSA. The improvement analyzed in CBSA-LB concerning CBSA is 27.81%. The total processing time improvement of the developed algorithm EMOSA_LB and EMOSAconcerning MOSA is 78.4% and 14.02%, and CBSA-LB concerning CBSA is 18%. Similarly, the optimization of total processing cost in developedEMOSA_LB and EMOSA is 92.8% and 49.72% over MOSA and 81.87% for CBSA-LB over CBSA. Also, in average response time, the improvement is 28.52% and 31.44% for EMOSA and EMOSA_LB over MOSA and 33.92% in CBSA-LB concerning CBSA. The percentage improvement of throughput is 60.45% and 96.6% in EMOSA and EMOSA_LB concerning MOSA, while CBSA_LB is 60.24 % concerning CBSA. Performance optimization of all other parameters concerning MOSA and CBSA is shown in Table 1 and Table 2. The improvement % age results show that EMOSA, EMOSA_LB,CBSA outperforms all the QoS values concerning MOSA and CBSA algorithm.

5. Conclusion

In this paper,many algorithms have been studied to improve the QoS of cloud networks with their pros and cons. The comparative analysis of different developed algorithms with the already existing algorithms in the literature has been done. The improvement percentage performance parameter calculation concerning base algorithms has been carried out. The calculations showed that with the new advancements in cloud computing, the optimization factor is getting improved. The outcome shows better performance over the existing base algorithms and is termed as the optimization comparison analysis. This performance analysis of the system targets various factors like availability, resource utilization, scalability, and performance gained by different QoS parameters on the cloud network. The broader descriptions of QoS are possible like security, Fault tolerance with various computing like green computing, fog computing, edge computing, in the field of cloud.

6. Acknowledgement

The authors are sincerely thankful to the Vice-Chancellor and Registrar of Maharshi Dayanand University, Rohtak, to pursue the Doctorate of Research.

References

1. P Rimal,E Choi, and I Lumb.A taxonomy and survey of cloud Computing systems.NCM 5th International Joint Conference INC,IMS,IDC.2009;44-51.

2. V. Lakra and D. K.Yadav.Multiobjective tasks scheduling algorithm for cloud computing throughput optimization, Procedia Comput. Sci..2015;48(C),107–113.
3. S.Potluri, K.S Rao. Optimization model for QoS based task scheduling in cloud computing environment.Indonesian Journal of Electrical Engineering and Computer Science.2020;18(2),1081-1088.
4. D. Ardagna, g.Casale, M.Ciavotta, J.FPérez and W.Wang,Quality-of-service in cloud computing:modeling techniques and their applications.Journal of Internet Services and Applications.2014;1-17.
5. O.Honeymol and S.Varghese.Cost Optimization With Increasing Qos In CloudComputing Environment.International Journal of Advances in Computer Science and Cloud Computing.2014;2(2).
6. A.Narwal, S.Dhingra.Task scheduling algorithm using multiobjective functions for cloud computing environment.International journal of control theory and applications.2017; 10(14).
7. S.Potluri, K.S Rao. Quality of Service based Task Scheduling Algorithms in Cloud Computing.International Journal of Electrical and Computer Engineering.2017; 7(2), 1088-1095.
8. A.Narwal, S.Dhingra.Enhanced Task Scheduling Algorithm Using Multiobjective Function for Cloud Computing Framework.P.Bhattacharyya et al.(Eds.):NGCT.2017;CCIS 827,110-121.
9. A.Thomas, G.Krishnalal, V.P Raj Jagathy.Credit Based Scheduling Algorithm in Cloud Computing Environment. Procedia Computer Science.2015; 46 ,913 – 920.
10. K.R Remesh Babu, A. A Joy, P. Samuel.Load Balancing of Tasks In Cloud ComputingEnvironment Based On Bee Colony Algorithm.International Conference on Advances in Computingand Communications.2015; 89-93.
11. A.Narwal, S.Dhingra.Performance analysis of multi objective algorithms for cloud computing framework.Journal of Adv. Res. Dyn. Control Syst.2019;11(5) ,2093–2099.
12. A.Narwal,S.Dhingra.Credit Based Scheduling with Load Balancing in Cloud Environment.International Journal of Advanced Trends in Computer Science and Engineering.2020; 9 (2).
13. Z.Qu, Y.Wang, L.Sun ,D.Peng, and Z.Li.Study QoS Optimization and Energy Saving Techniques in Cloud.Fog, Edge, and IoTHindawi Complexity.2020;Article ID 8964165, 1-16.
14. A.Hussain, M.Aleem, A.K,Muhammad Azhar Iqbal,M.A Islam,RALBA: A computation-aware load balancing scheduler for cloudcomputing.Cluster ComputingSpringer.2018
15. A.Narwal,S.Dhingra.Load Balancing using Enhanced Multi-Objective with Bee Colony Optimization in Cloud Networks.Pertanika Journal of Science and Technology.2020;28 (3),1049-1061..