Volume 8, No. 5, May-June 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Efficient Load Balancing Task Scheduling in Cloud Computing using Raven Roosting Optimization Algorithm

Ekta Rani
Research Scholar
Department of Computer Engineering
Punjabi University, Patiala
Punjab, India.

Harpreet Kaur
Assistant Professor
Department of Computer Engineering
Punjabi University, Patiala
Punjab, India.

Abstract: In this paper, a Raven Roosting Optimization Algorithm (RRO) is followed to light on the load balancing for task scheduling problems solution in cloud environment. Heterogeneity of birds, insects enroll in roosting. In raven Roosting, Roosts are information centers or can say servers and scrounge feature of common ravens inspired to solve problems. This technique is good enough to handle number of overloaded tasks transfer on Virtual Machines (VMs) by determining the availability of VMs capacity. Raven Roosting Optimization (RRO) random allocation of VMs to Cloudlets results huge change in makespan with respect to VM to which allocated. There is the possibility that simulation results shows better makespan, average response time, average waiting time, number of tasks migrated through Raven Roosting Optimization Algorithm.

Keywords: Cloud Computing, Load Balancing, Raven Roosting Optimization Algorithm (RRO), Task Scheduling.

I. INTRODUCTION

Dynamic extensibility, on demand, remotely access of services make cloud computing on fast track today's technology. For best performance we require effective scheduling of jobs thus task scheduling performed in cloud computing. Efficient Task scheduling is fulfillment of objective function that can be completion of tasks in less response time, less makespan, less cost, make resource utilization such that constraints are need to fulfill for solving many scheduling problems like load balancing, energy efficiency.

There are numerous conventional approaches of task scheduling, Min-Min, Min- Max, Genetic Algorithm (Agarwal & Srivastava, 2016), Honey Bee Algorithm (L.D & Krishna, 2013), and Round Robin, Particle Swarm optimization (Ramezani, Lu, & Hussain, 2014), Ant Colony Algorithm and many more. These strategies perform good but with some drawbacks. To defeat the scheduling issues and because of restriction of techniques, we have implemented a new soft computing technique that is Raven Roosting Optimization Technique (RRO). (Patel & Chawda, 2016)

In this work it is used for dealing with the complications of balancing load in cloud computing environment. Basically, raven roosting (O'Neill, Cui, & Brabazon, 2015) is mechanism of finding the food by creature so this method once place occupied by ravens, confirming the availability of sufficient for completion of their demand if not they start of seeking for food and flies on other location that can fulfill their need. Similarly this design is used for migration of load from one resource to another for efficient consummation of resource. This paper considers the makespan, response Time, Waiting Time and number of tasks need migrations in load balancing. Remaining organization of paper is section II discussing the previous work done in scheduling of tasks on resources that is Virtual Machines (VMs) in Cloud Computing. In section III, we draws the flow chart of RRO, according the steps are taken. And secondly the fitness equations are written which is used for calculation of load of cloudlets and the availability of Virtual Machines to handle cloudlets. Experiments and Results are shown in tables in section IV. At last the conclusion of paper and future work is mentioned.

II. RELATED WORK

(Patel & Chawda, 2016) Inspired from Nature many real world problem solving soft computing techniques came. This paper lists the latest soft computing techniques from foraging behavior of animals where Raven Roosting Optimization mentions for future research based on problem to be solved. (O'Neill, Cui, & Brabazon, 2015) A Social foraging of Common raven's behavior influenced the solution of problems in real world. Common ravens are considered as our task to solve this roosting sites are searched. In RRO best site recruit number of followers for completion. (Xin, 2016) Ant Colony resource allocation algorithm is implemented and results better as compared with Simulated Annealing and Genetic Algorithm. (Bei & Jun, 2016) Multi-population Genetic Algorithm (MPGA) is used for load balancing in cloud computing task scheduling. Simulation results less completion time and processing cost with balancing of load on resources in comparison of Adaptive Genetic Algorithm (AGA). For independent and dependent tasks, dynamic working can take in future work.

(Patel, Patel, Patel, & Patel, 2016) Improved Genetic Algorithm used for reduction in load balancing improves resource utilization and reduces response time. This paper focus on the Virtual Machines that are capable to handle but in future more methods can use for selection of Virtual Machines keeping the amount of load under consideration. (Agarwal & Srivastava, 2016) Genetic Algorithm (GA) is implemented with Service Level agreement (SLAs) conditions for better Quality of Service (QoS) and this discussed the future work on GA by different QoS. (Mittal & katal, 2016) which accumulates best results in distribution and scalability of environment of cloudlet scheduling. (kaur & sharma, 2016) This depicts the results of using enhanced optimal cost scheduling for load balancing on VMs make profitable completion and maximize resource utilization with later research on power and CPU usage possible by using this proposed algorithm. (Dasgupta, Mandal, Dutta, Mondal, & Dam, 2013) In this GA observed with Stochastic Hill Climbing (SHC), FCFs and RR exceeds in performance. It also evaluates time complexity and response time with different configuration of cloud and by variety of VMs. (Vanithaa & Marikkannu, 2017) this paper purposed a dynamic well organized load balancing algorithm allows large number of servers connected with Virtual Machines with reducing the power consumption in cloud computing. (Singh, Abraham, & Narayana lyengar, 2016) This presents the various min-min modified algorithms like load balancing improved min-min algorithm (LBIMM), User Priority Guided Load Balancing Improved Min-Min (PA-LBIMM), Enhanced Load Balancing Improved Min-Min (ELBIMM), Min-Min Ant Colony Algorithm (MMAC) and gives attention on the traditional Min-Min's simplest and improvised use for optimized performance. (Ojasvee & Banyal, 2016) The purposed algorithm works in positive way for distribution of tasks by recursively checking the availability of hosts. (Namboothiri & Samuel Raj, 2016) this observes the problems in load balancing scheduling by comparing algorithms Round Robin, Throttled Load balancing, Join idle Queue on different parameters reliability, availability and resource utilization and indicates the further work can be done with Join Idle on different threshold values and for less cost in scheduling. (Mondal, Dasgupta, & Dutta, 2012) Local search Stochastic Hill Climbing Algorithm is implemented results better than First Come First Serve and Round Robin Algorithm in response time by taking different number of data centers. (Ariharan & Manakattu, 2015) Research work done on calculating the probability of neighbor resources then from these lightly loaded assigned, this strategy of choosing make more explorer from nodes to nodes improves awareness to allocate efficient resource.

III. LOAD BALANCING TASK SCHEDULING USING RAVNE ROOSTING OPTIMIZATION ALGORITHM

In Raven Roosting Optimization, fitness value is calculated which instantly effects the performance of algorithm for the sake of load balancing task scheduling solutions in cloud environment.

A. Fitness Function

Raven Roosting Load balancing Optimization algorithm balance the load of cloudlets to VMs by calculating Load of cloudlets on VM.

Load of Cloudlet on a VM is

$$L_c = (w_1 \times NIC/MIPS) + (w_2 \times a)$$

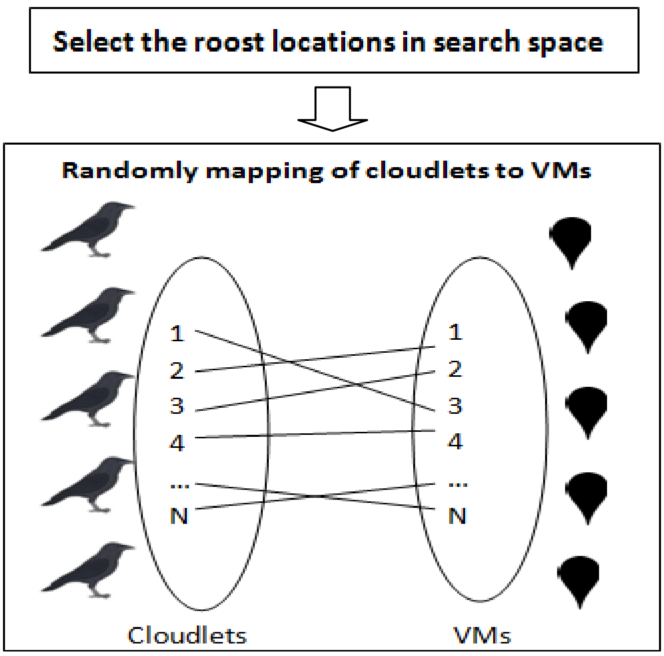
L_c is load of cloudlet on VM to which cloudlet is assigned. NIC is Number of Instructions of cloudlets MIPs is Million Instruction Per Second. 'a' is waiting delay of cloudlet. $w_1 and\ w_2$ are two weights values used to find load. More weight given to factor according to user's preference or importance given according to particular aspect. (Dasgupta, Mandal, Dutta, Mondal, & Dam, 2013)

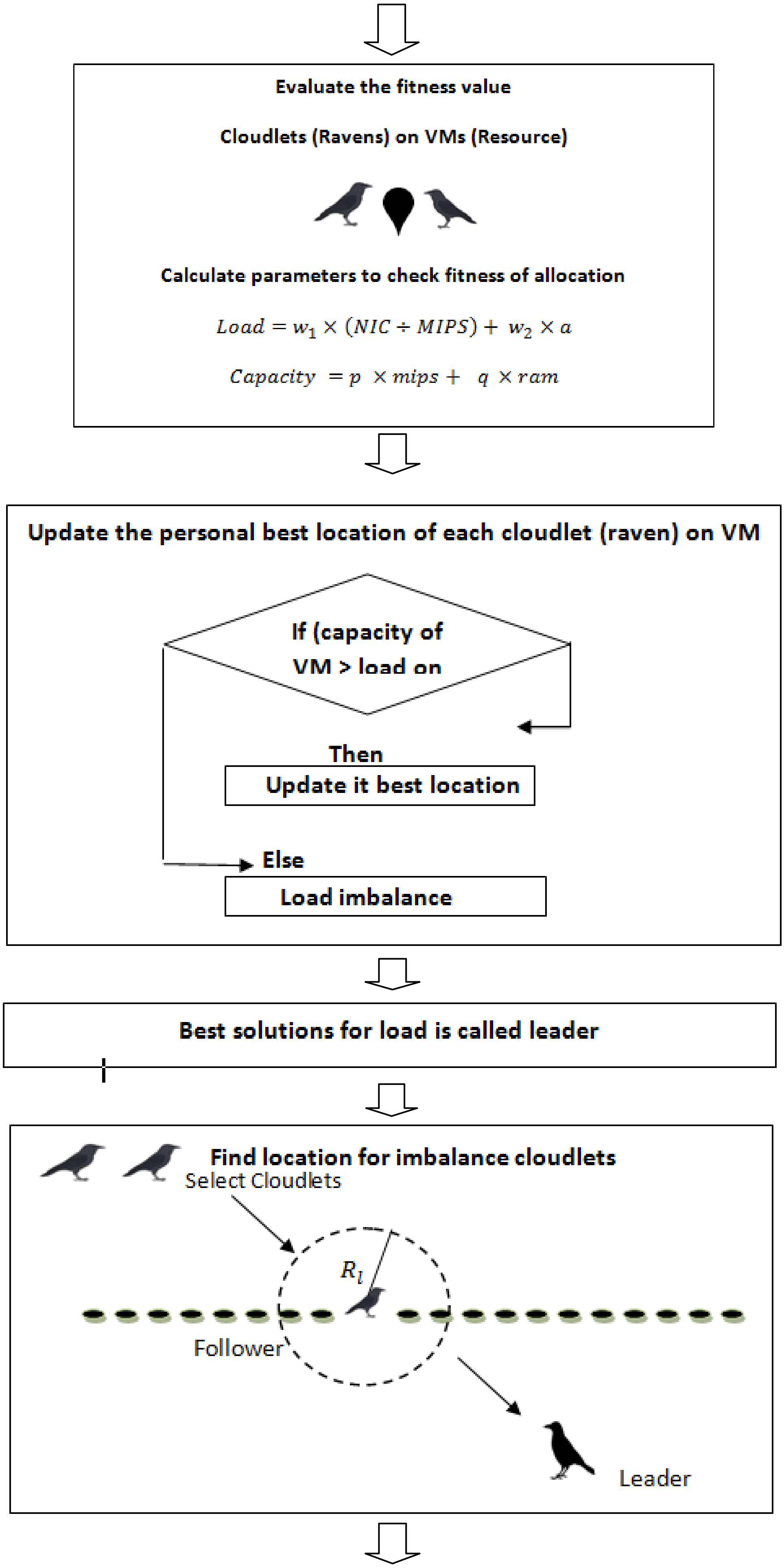
Capacity of a VM is

$$C_{VM} = p \times \text{mips} + \text{q} \times \text{ram}$$

MIPS = Million Instruction per Second of VM, RAM = ram of Virtual Memory. P and q are probability factor given to VM parameters.

B. Raven Roosting Optimization Algorithm




Evaluate the fitness value
Cloudlets (Ravens) on VMs (Resource)
Calculate parameters to check fitness of allocation
$Load = w_1 \times (NIC \div MIPS) + w_2 \times a$
$Capacity = p \times mips + q \times ram$
Update the personal best location of each cloudlet (raven) on VM
If (capacity of
VM > load on
Then
Update it best location
Else
Load imbalance
Best solutions for load is called leader
Find location for imbalance cloudlets
Select Cloudlets
R_l
Follower
Leader

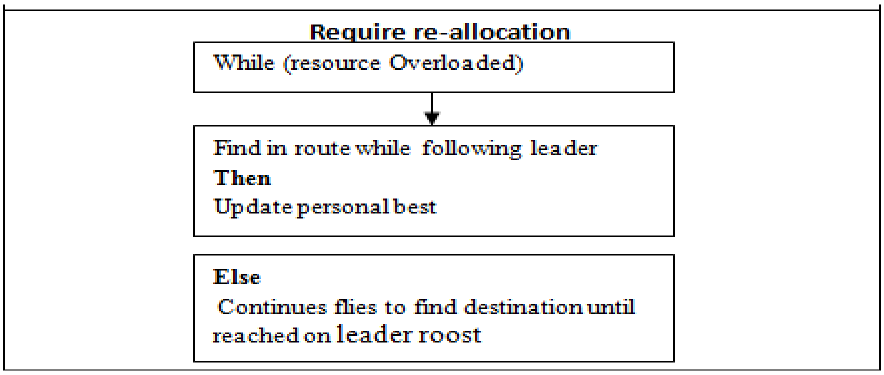


Figure 1: Flow Chart of Raven Roosting Optimization

Figure 1 describes the RRO (O'Neill, Cui, & Brabazon, 2015) working in solving load balancing problem of scheduling. Initially, ravens that are cloudlets submitted to roosting locations indicate our resources (Virtual Machines). Using fitness function, best location of each cloudlet is determined by calculating load correspondence to Virtual Machines on which task aligned to finish work. Condition is applied such that if load is imbalance according to ravens foraging behavior of recruiting followers cloudlets are migrated to different VMs to balance load.
There is two ways to find resource (Virtual Machine) with respect to leader (location of best solution), one is in the route if enough capacity of VM present to handle load cloudlet is allocated that specific VM. Otherwise leader become destination of this cloudlet.

IV. EVALAUTION OF RESULTS

Cloud computing system has to handle different load on every machine so in this implementation, execution of different number of cloudlets on different VMs is done by randomly allocation under simulation environment of cloud. Average Waiting Time (AVT), Average Response Time (ART), makespan and degree of load imbalance these parameters are calculated by considering different length of cloudlets on different MIPS (Million Instruction per Second) and RAM of VMs.

Table 1: Average waiting Time (AWT), Average Response Time (ART) and Number of Task migrations of different tasks on different VMs

S. No	No. of Cloudlets	No. of VMs	AWT	ART	Number of tasks Migrated
1	4	4	0.09	0.1	1
2	8	8	0.10	0.09	3
3	8	5	0.09	0.09	5
4	12	10	0.09	0.09	4
5	18	18	0.10	0.10	6

Table 2: Makespan of 4 Cloudlets to 4 VMs of different characteristics

S. No.	Cloudlet No.	VM	Makespan (in seconds)
1	0	3	333
2	1	2	437
3	2	3	777
4	3	0	1250

Table 1 drawn above shows the experiments undertaken by assigning different input values of Number of Cloudlets and Virtual Machines. Number of cloudlets randomly mapped to VMs in such a way that it is not necessary that each cloudlet assigned on different VM, it may happen that two or more cloudlets assigned on same VM. This type of random allocation effect the system by overload the Virtual Machines where we need to migrate cloudlets to another Virtual Machines for efficient working.

Table 2 represent the first experiment with 4 cloudlets of different length thus this different load of cloudlets on VMs completed with different makespan according to Capacity of VM.

Both Table 3 and Table 4 display the list of 8 cloudlets assignment but in Table 3 same 8 VMs are used and in Table 4 only 5 VMs are used for task completion. Table 5 presents the 12 cloudlets makespan on same number of VMs.

Table 3: Allocation of 8 Cloudlets to 8 VMs

S. No.	Cloudlet No.	VM	Makespan
1	0	7	230.77
2	1	7	384.62
3	2	3	611.11
4	3	6	971.67
5	4	4	1150.0
6	5	5	1363.64
7	6	5	19909.0
8	7	2	1937.54

Table 4: Assignment of 8 tasks to 5 VMs

S. No.	Cloudlet No.	VM	Makespan (in seconds)
1	0	3	166
2	1	4	700
3	2	4	1100
4	3	0	1571.43
5	4	0	2714
6	5	0	3562.5
7	6	0	4062
8	7	0	4562

Table 5: Correspondence of 12 tasks with 10 VMs

S. No.	Cloudlet No.	VM	Makespan (in seconds)
1	0	6	125
2	1	3	777.78
3	2	0	916.67
4	3	3	1444.44
5	4	8	1928.57
6	5	5	2045.45
7	6	9	2066
8	7	8	2214
9	8	9	2466.67
10	9	5	2772.73
11	10	1	2785.71
12	11	5	3863.63

V. EVALAUTION OF RESULTS

Starting from less number of cloudlets, then keep increase in cloudlets and Virtual Machines allocation results are taken. Not always same in number but also by taking more cloudlets and less Virtual Machines experiments attempted which results in changes in Migrations. Figure 2 represents that once the allocation is completed this end not always the boost the migration. By using RRO, it is analyzed from graph that when 8 Cloudlets are executed on 5 Machines this required 5 migrations to balance the load. After this 12 cloudlets transfer on 10 Virtual Machines demands 4 migrations which is less than previous one. Thus rise in cloudlets or more machines presence doesn't outcome with more migration.

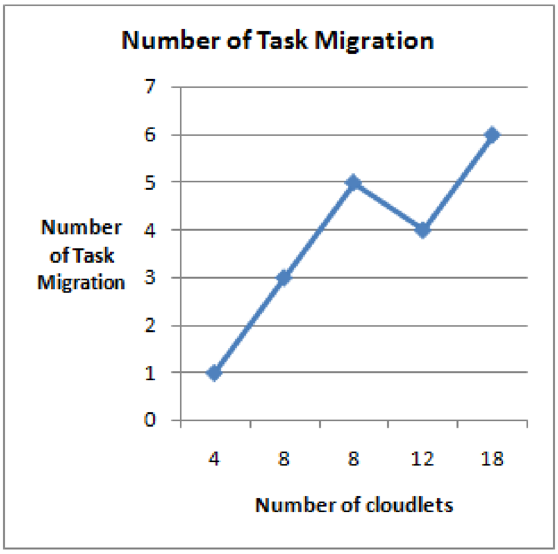


Figure 2: Number of Task Migration with increase in Cloudlets

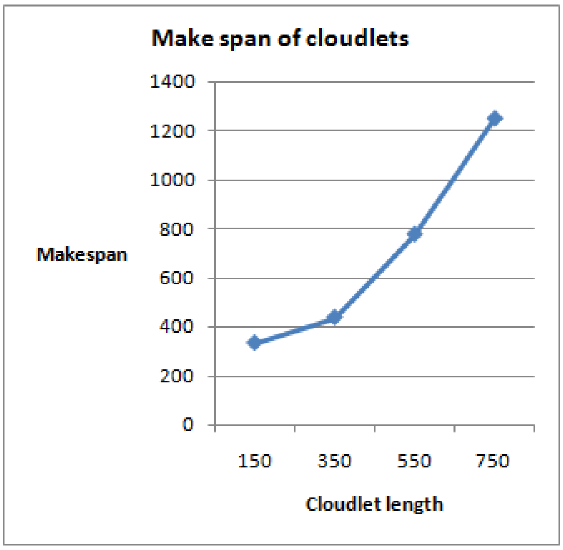


Figure 3: Makespan of cloudlets with different length cloudlets

Figure 3 illustrates the makespan on different length of cloudlets. While creating cloudlets 3 parameters that is input file size, output file size and length of cloudlets considered. Length of cloudlet is significant for efficient performance and after allotment of growth in cloudlet lengths makespan of cloudlets on VMs different. This experiment is also executed with same capacity and with different capacity of VMs. The increase occurs in makespan but this large value of makespan is because of Time shared policy used in cloudlets.

VI. CONCLUSION AND FUTURE SCOPE

This paper efforts in load balancing solution in cloud environment by using RRO. The hurried customer's increment on cloud makes rise in need of better load balancing scheduling. The dynamic behavior of this results not always cause more migrations with increase in cloudlets as analysis in figure 3. And the overload tasks migrated to search for VMs in roosting locations keeping the knowledge of ability of VMs which concluded decrease in execution time. From Analysis it is achieved that RRO performs well. In future, we can compare RRO with others implementations make it more optimized and calculate more parameter like cost of usage VMs.

REFERENCES

1. Agarwal, M., Srivastava, D.: A Genetic Algorithm inspired task scheduling in Cloud Computing. In : International Conference on Computing, Communication and Automation (ICCCA2016) (2016)
2. L.D, D., Krishna, P. V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing (2013)
3. Ramezani, F., Lu, j., Hussain, F.: Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization. International journal of Parallel Programming (2014)
4. Patel, J., Chawda, B.: Natural Computing Algorithms – A Survey. International Journal of Emerging Technology and Advanced Engineering 6(6) (june 2016)
5. O'Neill, M., Cui, W., Brabazon, A.: The Raven Roosting Optimization Algorithm. Soft Computing-Methodologies and Applications (2015)
6. Xin, G.: Ant Colony Optimization Computing Resource Allocation Algorithm Based on Cloud Computing Environment. (2016)
7. Bei, W., Jun, L.: Load balancing Task Scheduling based on multi Population Genetic Algorithm in cloud computing. In : 35th Chinese control Conference, China (2016)
8. Patel, R., Patel, S., Patel, D., Patel, T.: Improved GA Using Population Reduction Load Bancing in Cloud Computing. 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI) (2016)
9. Mittal, S., katal, A.: An Optimized Task Scheduling Algorithm in Cloud Computing. In : 2016 IEEE 6th International Conference on Advanced Computing (2016)
10. kaur, s., sharma, s.: Load Balancing in Cloud Computing with Enhanced Optimal Cost Scheduling Algorithm. Imperial Journal of Interdisciplinary Research (IJIR) (2016)
11. Dasgupta, K., Mandal, B., Dutta, P., Mondal, J., Dam, S.: A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013 (2013)
12. Vanithaa, M., Marikkannu, P.: Effective resource utilization in cloud environment through a dynamic well Organised load balancing algorithm for virtual machines. Computers and Electrical Engineering (2017)
13. Singh, S., Abraham, T., Narayana lyengar, S.: A Review: Different Improvised Min-Min Load Balancing Algorithm in Cloud Computing Environment. Journal of Computer and Mathematical Sciences, (2016)
14. Ojasvee, K., Banyal, R.: Analysis and Improvement of Load Balancing in Cloud Computing. 2016 International Conference on ICT in Business Industry & Government (ICTBIG) (2016)
15. Namboothiri, B., Samuel Raj, J.: A comparative study on Job Scheduling Algorithm Augmenting Load Balancing in cloud. In : 2016 second International Conference on scinence Technology Engineering and Management(ICONSTEM) (2016)
16. Mondal, B., Dasgupta, K., Dutta, P.: Load Balancig in Cloud Computing using Stochastic Hill Climbing-A soft Computing Appproach. ELEVIER (2012)
17. Ariharan, V., Manakattu, S.: Neighbour Aware Random Sampling (NARS) algorithm for load balancing in Cloud computing. 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (2015)

