Discrete Artificial Bee Colony Algorithm for Load Balancing in Cloud Computing Environment

Neha Thakkar[1], Rajender Nath[2]
[1]M.Tech Scholar, [2]Professor
[1,2]Department of Computer science and Applications, Kurukshetra University, Kurukshetra, India

Abstract

Load balancing is the main challenge in cloud computing environment.It is a process of reassigning the totalload to the individual nodes of the collective system of the facilitate networks so that no one virtual machine is overloaded and no one is under-loadedin order to improve the response time of the job with maximum throughput in the system.This paper improvises the existing ABC algorithm and proposes Discrete Artificial Bee Colony (DABC) algorithmas a load optimization algorithm byintroducing changes in the employee bee phase, onlooker bee phase and scout bee phase. The mutations operator is added into the onlooker phase to get the new best solutions. The discrete operators are used for positions updations in these three phases which improve its make-span and average resource utilizations. The load balancing module is also considered in this algorithm that if one resource is overloaded then the task is transferred to the under-loaded resource Theexperimental result have shown the proposed algorithm perform better than the existing Artificial Bee Colony(ABC) algorithm and other algorithms in term of some parameters like Makespan and Average Resource Utilizations Ratio.

Keywords - *Cloud Computing, Load balancing,Artificial Bee Colony, Discrete Artificial Bee Colony algorithm, Make-Span, Average Resource Utilizations*

I. INTRODUCTION

Cloud Computing(CC) is a computational paradigm developed from distributed computing model lies between grid computing and supercomputing. The CC provides many services that are classified into three categories: Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS).Cloud delivers the service into three models namely public cloud, private cloud andhybrid cloud, community cloud.

Load balancing is a process of reassigning the total load to the individual nodes of the collective system of the facilitate networks and resources to improve the response time of the job with maximum throughput in the system´.Load balancing is all about availability, scalability and performance of resources for critical web-based applications.There are two types of load balancing techniques:static and dynamic load balancing techniques. A load balancing technique ensures that each system in the network has same amount of work at any instant of time.The important things to consider while developing such algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes.This means neither any of them is overloaded and nor under-loaded.

The rest of the paper is organized as follows: Section II present the related work on Artificial Bee Colony algorithm(ABC). In section III, Discrete Artificial Bee Colony (DABC) algorithm with load balancing module is proposed,in section VI the performance analysis of the proposed algorithm is done,the final conclusion is given in section VI.

II. RELATED WORKS

B. Akay et al. [1] proposed the Artificial Bee Colony (ABC) optimizations algorithm for training feed-forward neural networks. They described the drawback of the traditional algorithms to getting stuck in local minima and computational complexity. They described the evolutionary algorithm to train the neural network to overcome the issue in traditional algorithm. The result shown that the ABC algorithmbetter trained the neural network because of its good exploration and exploitation capabilities. A. Bernardino et al. [2] utilized ABC to solve non-split Weighted Ring Arc-Loading Problem (WRALP). The main objective of the technique was to standardize the routing for each demand, to lower the maximum arc load. The performance of the algorithm was thus compared with three other techniques tabu search (TS), classical GA and a Local Search Probability Binary PSO (LS-PBPSO).The results shown that the proposed algorithm performed better than the existing algorithms. Bin et

al.[3]hybridized ABC with DEalgorithm to enhance the convergence attributes and prevent ABC from getting stuck in the local optima.The differential operator was used in the ABC algorithm and split the technique into two: EDABC if the differential operator was carried out by the employed bees or LDABC when carried out by the onlooker bees. This was done to enhance the diversity of the solutions in the population and the capability of global search with the differentialoperator since it obeyed uniform distribution and createcandidate solution position that could fully represent the search space. They compared the performance of EDABC and LDABC with ABC and both techniques outperformed ABC, whereas EDABC was better than the LDABC. Yuce Baris et al. [4] proposed Honey Bees inspired Optimization Method. They described an optimization algorithm called the Bees Algorithm(BA), inspired from the natural foraging behavior of honey bees, to find the optimal solution. The algorithm performed an exploitative neighborhood search combined with random explorative search.Ramya V. et al. [5] proposed an algorithm of honey bee as a load balancing algorithm. In this paper an algorithm was defined in which the tasks were removed from the overloaded virtual machine (VM) to under-loaded VM. There could be more than one VM which could accept a task. The task had to be allocated to a VM based on the QoS criteria called task priority. There was a fight for the VM among tasks. When this fight was over, the winning task was allocated to the respective VM found and the details were updated. If a task did not find a suitable VM, it went for delayed allocationmeanwhile;it startedlisten the information updates by other tasks (similar to honey bees listening various dances).Baby Anna [6] proposed a research as the Improved Honey Bee algorithm which was inspired by load balancing with position updation. Their modified bee foraging technique with position updation applies the knowledge of previous solution and mainly looks for a better solution. The position of VM was updated in an efficient manner for each iteration. If the number of iterations was less, then the convergence rate was high. Hence, a global optimal solution was found out by achieving fast convergence speed.The algorithm ignored the idle condition of a VM even after the completion of jobs assigned to it.Rathore Monika et al. [7] proposed the honey bee forage technique to guarantee that no VM kept idle. Once the VM was idle, the algorithmic rule could build multiple tries to steal jobs from a random VM.In This paper the honey bee behavior galvanized load balancing (HBB_LB), described. It ignored the idle condition of virtual machine then finishes up within the wastage of interval. Therefore the worldwide best answer couldn't be detected at intervals a quick span of some time. In this paper, accumulated honey bee forage technique with random stealing is used for task allocation and load leveling. Once tasks area unit assigned to the VMs, current load calculated. If the VM becomes overloaded the task was transferred to the neighborhood VM whose load value was below threshold. Their Proposed algorithm ensures performance of the system and avoid system imbalance. Bhavya V.V. et al. [8] proposed an Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing. In this paper the performance of the honeybee foraging algorithm based on throughput calculated for each virtual machine. The goal was to find the appropriate VM with the help of the throughput calculated earlier and to allocate the load–this leads to more efficient load balancing.

III. PROPOSED LOAD BALNCING ALGORITHM

A. Existing Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC)[9]is an optimization algorithm based on the intelligent behavior of honey bee foraging. This algorithm consists of four phases-Initialization of Swarm, Employed Bee, Onlooker Bee, and Scout Bee.

Initialization of Swarm:The ABC algorithm that mimics the extraordinary food foraging behaviorof real honey bees has three control parameters: First is the Population size (SN) ,the number of food sources (or solutions) in the population. The second is Maximum Cycle Number (MCN) refers to the maximum number of generations. The third parameter is Limit used to diversify the search, to determine the number of allowable generations for which each non-improved food source is to be abandoned. ABC generates a uniformly distributed population of SN solutions X_I (I = 1, 2 ...SN) is a D-dimensional vector. Here D is the no of variables in the optimization problem and X_I represents the I^{th} food source in the populations. Each food source is generated by using Eq.(1):

$X_{IJ}= X_{minJ} +rand[0,1] (X_{maxJ} - X_{minJ})$, $\forall_J$=1,2,…D (1)

Where, rand[0,1] is a function that randomly generated a regularly scattered arbitrary number in the range [0,1], where $X_{maxJ}$and$X_{minJ}$are bounds of X_I in J^{th} directions.

.

Employed Bee Phase:This is the second phase of the ABC algorithm. The position of current solutions modernized with the help of the knowledge of an individual's understanding and the appropriateness of the recently established solutions. Existing food sources replaced with the innovative food source having better value of fitness. The position of J^{th} dimension of I^{th} candidate modernizes using Eq. (2):

$V_{IJ}=X_{IJ}+Ø_{IJ}(X_{IJ}-X_{KJ})$ (2)

Where $Ø_{IJ}(X_{IJ}-X_{KJ})$ is identified as step size. In this formula J and k are two arbitrarily selected indices. Such that k ∈ {1, 2, ..., SN}, j ∈ {1, 2, ...,D} and k ≠ i make certain that there must be some significant enhancement in step size and $Ø_{IJ}$ is a random no. between [-1,1]

Onlooker Bee Phase: The third key phase of ABC algorithm is onlooker bee phase. The counting of onlooker bees is identical with the quantity of employed bees. For the period of this segment all employed bee share quality of novel food sources through onlooker bees in form of fitness. Every food source judged based on its probability of selection. There are various methods to compute the probability, generally it calculated in term of fitness. The highly fitted solution gets elected by the onlooker. It has to be a function of fitness. The fitness of each food source is determined by its function value if the function value is greater than zero then by using the (3) equations calculate the fitness value and if not calculate the fitness value by using the equations (4)

Fitness=1/2* FunctionValue+1(3)

Fitness=1+ fabs(1/FunctionValue)(4)

The selection of probability for each food source is determined by its fitness as per Eq. (5):

$P_{IJ}= fit_I/€Fit_{I(5)}$

Algorithm: Artificial Bee Colony Algorithm

Begin

1. Initialize all the essential parameters;
2. Reiterate till the criteria for termination do not meet
3. Employed bee phase for calculating new food sources.
4. Onlooker bees phase for position update of the food sources according to the quality of food sources.
5. Scout bee phase generate new food sources in place of discarded food sources.
6. Remember the most excellent food source recognized up to now.

End

Scout Bee Phase: The final phase of ABC algorithm is a scout bee phase. If the positions of a food source are not updated for a predetermined number of cycles, then the food source is assumed to be abandoned and scout bee phase is started. In this phase the bees associated with the abandoned food source becomes scout bee and the food source is replaced by the randomly chosen food source within the search space. New food sources generated using Eq. (6)

$X_{IJ}= X_{minJ} +rand[0,1] (X_{maxJ} - X_{minJ})$ (6)

As discussed in above four phases, it can be concluded that the ABC algorithm has three control parameters. That is the maximum number of cycles, the limit and SN (that is a number of food sources).

Problem statement: Many improvements[10][11][12] are proposed in the literature.For efficient load balancing schedule all the tasks are assigned to the virtual machines in such a way that cloud users can execute their tasks in minimum time and maximum resource utilizations. Thepositions updations inalgorithm [13] are given in equations (1)(2)(3)(4)(5)(6) are based on the real no's. The formulas are used to update the positions in employed bee, onlooker bee and the scout bee phase.The makespan and average resource utilization of the existing algorithm are still poor. So this paper intends to further improvise the algorithm in term of makespan and average resource utilization.

B. Proposed Discrete Artificial Bee Colony(DABC)

To address the problem stated in the problem statement there is the step added in the proposed DABC algorithm and the positions updations are done in the all three phases:-employed bee, onlooker bee, scout bee by using discrete operatorsvizflip,swap,slide.The additional mutation operator of Genetic Algorithm (GA) is used in the classical ABC algorithm for solving the job scheduling problem with the criterion to decrease the makespan and increase the resource utilization.The additional mutations step is added into the onlooker bee phase. Through mutation, on the one side, there is a chance of changing the local best position, and the algorithm may not be trapped into local optima. Selection of food source is done by the tournament selections. Food Source is selected arbitrarily from the food size and mutation is performed. In mutation, generated offspring's replaces the older offspring's. The mutation operator used in this paper is uniform mutation. Figure 1 shows the high level flow chart of the proposed algorithm. The algorithm considers the four phases:

Initialize Populations,EmployedBee, Onlooker bee, Scout Bee Phase.

Initialize Populations: The DABC algorithm that mimics the extraordinary food foraging behavior of real honey bees has three control parameters: First is the Population size (SN) , the number of food sources (or solutions) in the population. The second is Maximum Cycle Number (MCN) refers to the maximum number of generations. The third parameter is global best value. The random no of solutions are generated in the range of the no of resources and the no of tasks.

Employed Bee Phase: In Employed Bee phase the discrete operators are used to update the positions of the food sources. The swap operator is used to change theJ^{th}row of existing algorithm is replaced with the new

randomly generated populations. If the new generated fitness is better than the existing populations. Then assign the new populations to the existing populations.The employed bee generated the solutions for all populations.

Onlooker Bee Phase: The onlooker bees prefer the food source depending on nectar information's provided by employed bees. In Onlooker Bee phase the best four pair elements are chosen by the tournament selections.The best fitness value is kept constant and the other three are replaced by the discrete operators that are the slide, swap, flip operator. The bestnew populations are generated.

Scout Bee Phase: The final phase of DABC algorithm is scout bee phase.If the positions of a food source are not updated for a predetermined number of cycles, then the food source is assumed to be abandoned and scout bee phase is started. In scout bee phase the solutions which are less than the average fitness are considered as the abandoned and are replaced by the new randomly generated solutions.

The details descriptions of each phases is given below.

Flow Chart of ProposedAlgorithm

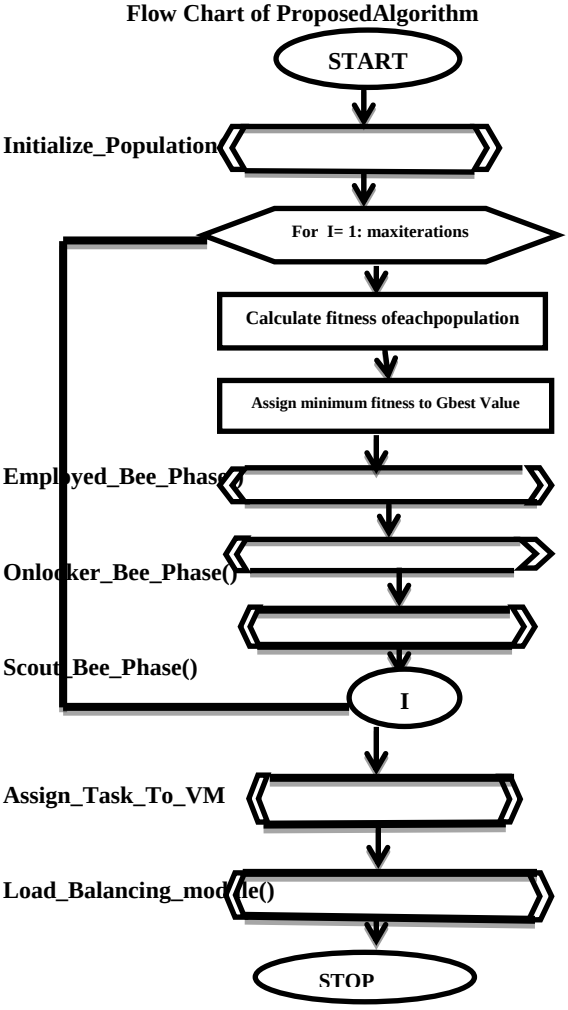

Figure 1. High Level Flow Chart of DABC algorithm.

Algorithm Initialize_Population()

Begin
//calculate executions time for each virtual machine.

1. For all submitted tasks in; Ti
   1.1. For all resources; Rj

   Et(i,j)=tasksize(i)/resourcespeed(j);
   1.2 End For;
2. End For;
3. Initialize population's size with some random value. //pop=40;
4. Generate pop no of random solutions. //The random functions are generating the solutions in the range of no of resources.
5. Initialize maximum iteration and global bestvalue. // initializes GbestValue infinity and maximum iterations 100.

End

Algorithm Employed_Bee_Phase()

Begin

1. For Each populations size
   1.1 Take the j_{th} row of each populations, which is randomly generated in first phase.

// existing populations

   1.2 A new population for this row is also generated randomly.

// new randomly generated populations.

   1.3 Generate any two random values and swap the positions of these values of new generated populations.
   1.4 Calculated the fitness of new generated populations.

// fitness in-term of the make-span.

   1.5 If the new generated fitness is less than the existing fitness than assign this row to existing populations and assign this fitness. // the fitness is considered in-term of make-span that's why we check the less fitness.
2. End For;

End

Algorithm Onlooker_Bee_Phase()

Begin

1. Generate a row vector by containing random permutations of the integer to the population's size.
2. For each population size pair the four elements.
   // increment of four.
   - 2.1 Pick the four elements from random order permutations and calculate their fitness. // calculate the make-span and store their value into a variable.
   - 2.2 Choose the minimum fitness value from these four elements.
   - 2.3 Store the id of the minimum best fitness populations.
   - 2.4 Multiply the no of task to the random no and sort them. // Ceil function is also applied on this to round the element to the greatest or equal integer and sort them in ascending order. There is the tournament selection applied.
   - 2.5 Cut into two points and store into the variable. // cut off apply for calculating the two numbers for which positions we can apply the operations.
   - 2.6 For mutate to get the best three routes which are selected further. Keep the best row exists and applies the discrete operators to other three rows like flip, swap and slide. And the new populations generated for these functions.
   - 2.7 End For;
3. End For; new populations assign to the old populations.

End

Algorithm Scout_Bee_Phase()

Begin

1. Calculate the average fitness of the population.
2. For each population size compare the solutions. The solutions which are abundant (whose value is less than the average fitness) are replaced by the random solutions.
3. End For;

End

Algorithm Assign_Task_To_VM()

Begin

1) For all submitted tasks in the set; Ti
   - 1.1. Assign each schedule value to each virtual machine.
   - 1.2. Assign the task on that virtual machine.
2) End For;
3) For all resources R_J
   - 3.1 Calculate selected time and selected task for the virtual machine.
4) End For
   //Resource Assignment
5) For all resources R_J
   - 5.1 Calculate the Ready time of each resource.
   - 5.2 Assign T_K to the resource R_J which gives minimum expected completions time.
   - 5.3 Remove TK from the tasks set
   - 5.4 Update ready time r_J for select R_j
6) End For

End

This algorithm generates the random solution and assigns the task to suitable virtual machines. The algorithm reduces the makespan and increases the average resource utilizations. It calculates the minimum make-span on each virtual machine and returns that virtual machine which has minimum makespan.

Algorithm Load_Balancing_Module()

//Rescheduling to balance the load

Begin

1. Do while the most heavy load resource is considered no need for rescheduling
2. Find task Ti that cost minimum execution time on the heavy load resource Rj
3. Find the minimum completion time of Ti produced by resource Rk
4. If such minimum completion time < make-span Reassign Task Ti to Resource Rk
5. Update the ready time of both Rj and Rk
6. End If
7. End Do

//where Make-span represents maximum completion time of all tasks which equals to the completion time of the most heavy load resource.

End

IV. PERFORMANCE ANYALYSIS OF PROPOSED DABC ALGORITHM

The performance of the proposed algorithm is compared with the existing load balancing algorithms likemin-min, max-min, existing ABC and the results are shown on the basis of parameters:- makespan, average resource utilizations ratio (ARUR), ready time (RT) of each resource.

- **Makespan:** Makespan is a measure of the throughput of the heterogeneous cloud system. It can be calculated as the following relation:
  Makespan=max (rt_j)

  Where rt_j denotes the ready time of each resource after scheduled. The less the makespan of a scheduling algorithm the better it works.

- **Average Resource Utilization Ratio (ARUR):** ARUR is calculated through the following relation:
  ARUR= mean (rt_j)/Makespan*100

- **Ready Time(RT):**Ready time is the amount of time that a resource is ready to process a job but is waiting on a scheduler for access to a physical processor.

- **Meta-Task(MT):**It is a set of tasks, which are to be mapped on available resources.
  MT={t_1,t_2,t_3,……., t_n)

Where t_1,t_2,t_3…t_n is the n no of tasks.

The proposed and existing algorithmsare implemented in MATLAB and the experiments performed on the data are given in Table I and Table II. Table I, represents the processing speed of each resource while Table II, represents the task size.

Resources	Processing Speed(MB/Sec)
R1	10
R2	8
R3	9
R4	5
R5	3

Table I. Resources Specification

Tasks	Processing Speed(MB/Sec)
T1	95
T2	24
T3	61
T4	49
T5	89
T6	76
T7	46
T8	3
T9	82
T10	45

Table II. Tasks Specification

The result is calculated two times of the DABC algorithm. First the best solutions after the three phases is calculated and then the second time the result is calculated after the load balancing module (DABCLB).The result of make-span is shown with the help of graph in Figure2.

Figure 2: The graph represents the makespan of the proposed algorithm and the existing algorithm. The proposed algorithm gives the better result as compared to the others algorithms.

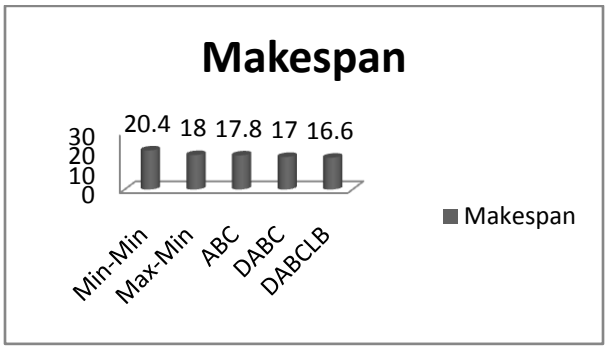

Figure 2: Performance Analysis Using Makespan.

Figure 3: The graph represents the average resource utilization ratio of the proposed algorithm and the existing algorithm. The average resource utilization ratio (ARUR) is better in the DABC and DABCLB algorithms as compare to existing algorithms.

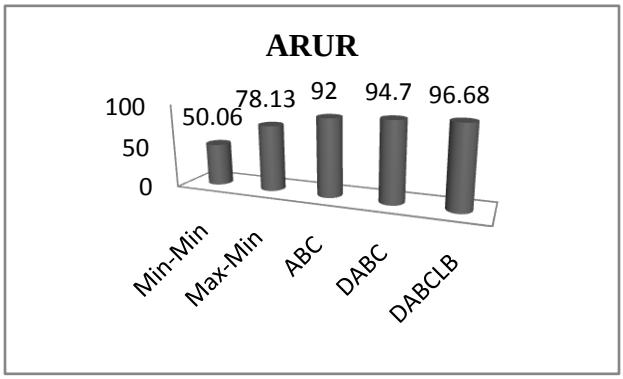

Figure 3: Performance Analysis Using Average Resource Utilizations.

Table III: The graph represents the Ready time on each resource of the proposed algorithm and the existing algorithm.

Approach	R1	R2	R3	R4	R5
Min-Min	20.4	16	18	15.2	0
Max-Min	18	15.87	15.66	15.2	15.33
ABC	16.7	17.5	13.88	17.8	16.33
DABC	16.5	15.62	17	16.4	15
DABCLB	16.5	16	16.66	16.4	15

Table III: Ready time of each algorithm on each resource.

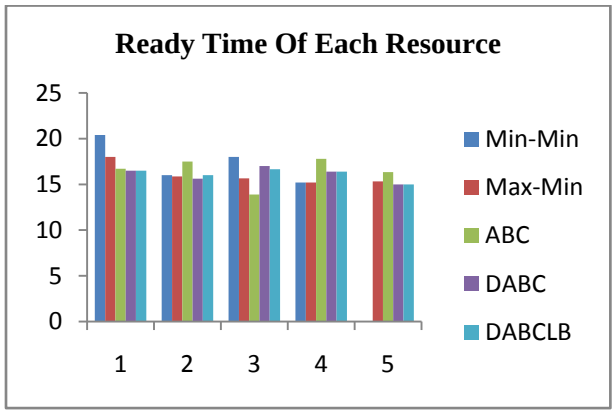

Figure 5: Ready Time on Each Virtual machine

As can be seen from Figure 3,4,5, compare with min-min, max-min, ABC, both DABC and DABCLB decreases the makespan, and increase the average resource utilizations ratio (ARUR).The DABC and DABCLB does a great job to reduce the ready time of each resource as compared to the others algorithms.

V. CONCULSION

This paper has proposed the Discrete Artificial Bee Colony (DABC) algorithm for load balancing in CC environment. The limitations of the existing algorithm has been poor makespan and average resource utilization. To further improve the performance of the existing algorithm the discrete formulas are used to update the positions in the proposed algorithm. In this algorithm additional mutation and crossover operator of Genetic Algorithm (GA) is used in the classical ABC algorithm for solving the job scheduling problem.The DABC algorithm has been evaluated experimentally and found that it improves the makespan, average resource utilizations ratio(ARUR), and ready time of each resource. To show the efficiency of the proposed DABC algorithmcompared with existing algorithm. The experimental results shows that the formula is effective when put next with existing algorithms. The approach illustrates that there's a big improvement in make-span and average resourceutilizations ratio(ARUR), average execution time and reduction in waiting time of tasks on queue.

.

REFERENCE

[1] D.Karaboga, B. Akay, and C. Ozturk,"Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks" Modeling decisions for artificial intelligence, pages 318–329, 2007.

[2] A.Bernardino, E. Bernardino, J. Sa ´nchezPe ´rez, J. Go ´mez-Pulido, and M. VegaRodr´ıguez, "Efficient load balancing for a resilient packet ring using artificial bee colony" Applications of Evolutionary Computation, pages 61–70, 2010.

[3] W.Bin et al.,"Differential artificial bee colony algorithm for global numerical optimization" Journal of Computers, 6(5):841-848, 2011.

[4] Baris Yuce, Michael S. Packianather, Ernesto Mastrocinque , Duc Truong Pham and Alfredo Lambiase , "Honey Bees Inspired Optimization Method: The Bees Algorithm," in the proc. of insects ISSN 2075-4450, 2013.

[5] V.Ramya, S. Ranjitha, A. Sathya Sofia and P. Ganesh Kumar, "Load Balancing of Tasks in Cloud Computing Environment Using Honey Bee Behavior," in the proc. of International Journal of System Design and Information Processing, Vol. 2, No. 2, June 2014.

[6] Ms.Anna Baby ," Improved Honey Bee inspired load balancing of tasks with position updation" ,in the proc. of International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 3 Issue IV, April 2015.

[7] Monika Rathore, Sarvesh Rai, Navdeep Saluja ," Randomized Honey Bee Load Balancing Algorithm in Cloud Computing System " ,in the proc. Of (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.7,2016.

[8] V.V.Bhavya, K.P. Rejina and A.S. Mahesh," An Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing",in the proc. Of International Journal of Pure and Applied Mathematics, Volume 114 No. 11 2017.

[9] D.Karaboga. An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.[10] Deepika Nee Miku,Preeti gulia,"Improve Performance of Load Balancing using Artificial Bee Colony in Grid Computing",International Journal of Computer Applications (0975 – 8887) Volume 86 – No 14, January 2014

[10] Warangkhana Kimpan, Member, IEEE,"Heuristic Task Scheduling with Artificial Bee Colony Algorithm for Virtual Machines" in the proc. Of 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems.

[11] WeiMao, Heng-youLan, Hao-ruLi, "A New Modified Artificial Bee Colony Algorithm with Exponential Function Adaptive Steps",Hindawi Publishing Corporation Computational Intelligence and Neuroscience Volume 2016, Article ID 9820294, 13 pages.

[12] Sangeeta sharma,pawan Bhambu, " Artificial Bee Colony Algorithm: A survey International journal of computer application, volume 149-No.4,September 2016.