Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **1**, January **2017**

A New Scheduling Algorithm Based on Ant Colony Algorithm and Cloud Load Balancing

Linlin Tang[a], Kaiqiang Ma, Zuohua Li[b]

Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
[a]hittang@126.com, [b]lizuohua@hit.edu.cn

Received June, 2016; revised October, 2016

Abstract. *A good task scheduling algorithm can effectively coordinate and allocate virtual machine resources, effectively reduce the total execution time and resource consumption. so as to achieve the best performance of a cloud system. In this paper, the cloud computing task scheduling is considered to be based on the load balance degree, and the load balancing factor LBF(LOAD BALANCE FACTOR) is proposed according to the load balance degree of the virtual machine scheduling in real time. The update rule of pheromone is optimized. This algorithm has been carried out a lot of Experiments on the CloudSim, which have been given in our work and the good results show its efficiency.*
Keywords: Cloud computing; Load balancing; Ant colony algorithm; Intelligent computation.

1. **Introduction.** Cloud computing is a new technology, which is developed with the distributed computing, parallel computing and grid computing techniques.Cloud task scheduling is an NP-complete problem in general [7].Virtual technology in cloud computing is the use of the hardware resources of the data center as the virtual resource pool for the unified management and external services. By doing so, the so called pay-as-you-go service is provided. Resource is transparent to the user in business service model. For the users task processing time is directly linked to the cost of the user to pay, the task processing time span becomes an important issue to care. Considering the amount of tasks and the massive data , so how to provide a reasonable allocation of virtual machine resources and how to arrange the tasks of the user efficiently is very important for a whole system. At the same time, the load of the system need to be maintained in a relatively balanced state. Which are the key and difficult points in the research of cloud computing.The existing cloud computing task scheduling algorithm is relatively simple: from the user's point of view, the existing cloud computing task scheduling algorithms mostly focus on shortening the time or only considering the response speed[1, 10]. From the view of cloud resource service providers (system resource perspective), the existing scheduling algorithms and allocation strategy pay more attention to economic efficiency and resource utilization[2, 12].the task scheduling algorithm, can shorten the task execution time, but it often causes the system in non-balanced state. The utilization of some resources in the resource pool is high, and utilization of other resources is very low.

Cloud computing service model is required to meet the needs of a wide range of users, which makes the task scheduling of cloud computing complex, in full consideration the different needs of users and also meet the needs of the system load balancing.In the face of these new requirements, based on the load balancing and ant colony algorithm a new task scheduling strategy is designed in this paper.

2. **Related work.** According to the different cloud computing environments, QoS requirements of the user task scheduling are also different. On the one hand, large number of tasks and resources, task scheduling problem is extremely complex. On the other hand, the task scheduling strategy plays an important role in the performance of the cloud computing system. So in recent years, cloud computing task scheduling problem has attracted a large number of experts and scholars to carry out in-depth study. M. Aruna, D. Bhanu, R. Punithagowri in reference[11] proposed Weighted Round Robin. The proposed algorithm weights according to the VM's ability of handling the assignment. In one word, the better performance of the VM, the more tasks are submitted to it. The advantage is to further improve the resources utilization. But it still did not consider the completion time of a single request, the user experience is poor. The improved the algorithm based on honey bee behavior reducing the task execution time[6, 8], but they do not take the load balance of the system into consideration. Huankai Chen, Professor Frank Wang in User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing proposed improved scheduling Algorithm Min-Min algorithm[3]. It starts with a set of all unmapped tasks. Then the resource which has the minimum completion time for all tasks is found. Next, the task with the minimum size is selected and assigned to the corresponding resource (hence the name Min-Min)[4]. At last, the task is removed from set and the same procedure is repeated by Min-Min until all tasks are assigned .The disadvantage of this algorithm is that it does not consider the user's priority and fails to utilize the resources efficiently which lead to a load imbalance. Based on this point, Huankai Chen, Professor Frank Wang in reference[5] introduce the concept of users priority. We find that the scheduling algorithm can be seen as a tradeoff between the total time and user priority. The problem is that the algorithm in large-scale data is not good.The existing improved ant colony algorithm can reduce the task executing time, but has no obvious improvement to the load banlance. And the actual operation of the heuristic function is not specific and effective, resulting in the slow convergence of the whole optimization process. At the same time, it is easy to fall into local optimal solution, and it can't make effective adjustment according to the real-time situation of optimization in the random and deterministic. Generally speaking, the convergence speed and the accuracy of the optimization need to be improved.

Based on the classical ant colony algorithm, a new system has been proposed in this paper, they can not only accelerate the update of pheromone, but also can increase the ability of global search. By introducing the LBF load balancing real-time factor, the entire system maintains a good degree of load balancing.

3. **The description for cloud computing task scheduling problem.** The task scheduling algorithm has two main purposes. First, it is used for the user to transparently submit the task to the resource; second, it can also be used to increase the utilization rate of resource and to meet the requirement of the user. The main work for scheduling is to make the task accomplished efficiently, and the specific formulas in this process can be shown as follows:

$$t_{ij} = \frac{imputSize_i + outputSize_i}{bandwidth_j} \tag{1}$$

$$e_{ij} = \frac{Length_i}{Mips_j} \tag{2}$$

$$f_{ij} = t_{ij} + e_{ij} \tag{3}$$

In the formula (1),$inputSize_i$ and $outSize_i$ represent the input size and output size of the $task_i$. $bandwidth_j$ means the bandwidth capacity of the virtual machine, t_{ij} represents the transmission time of $task_i$ to vm_j ;In the formula (2), $Length_i$ represents the amount of computation of the $task_i$, $Mips_j$ represents the computing power of virtual machine whose number is j, e_{ij} is execution time; In the formula (3) , f_{ij} represents the total time that from submission to complete the single task.

4. Our proposed method.

4.1. Initialization of heuristic function and initialization of pheromone.

For the core of the whole ant colony scheduling algorithm, the heuristic function and the pheromone function are basis of the mutual competition and the interaction between the deterministic and stochastic of the whole swarm intelligence algorithm. In this paper, we use the main idea of Min-min algorithm, and introduce the advantages of its simple and reliable total completion time, adding load balancing mechanism to avoid the inherent drawbacks of Min-min, and use the results to initialize the heuristic function.

$$C_{ij} = \frac{1}{e_{ij}} \tag{4}$$

$$C_{ij} = \frac{1}{e_{ij}} + X \tag{5}$$

$$\tau_{ij} = \frac{Mips}{A} + \frac{Bandwindths}{B} \tag{6}$$

The upper (4) (5) are used to update the C_{ij} heuristic function. If it is the result of Min-min scheduling algorithm use (5), else (4), (6) τ_{ij} initialization function of pheromone which is used to measure the overall performance of a virtual machine, τ_{ij} represents the paths pheromone of $task_i$ to vm_j .

4.2. Virtual node selection.

After determining the heuristic function and the pheromone function, the probability of virtual machine to be scheduled can be calculated according to the following formula (7):

$$P_{ij}(t) = \frac{[C_{ij}(t)]^a * [\tau_{ij}(t)]^b}{\sum_{x=1}^{m}[C_{ix}(t)]^a * [\tau_{ix}(t)]^b} \tag{7}$$

In the above model, a is expected to show the importance of visibility, which reflects the weight of the ant in the task scheduling, and b is the pheromone heuristic factor, which reflects the influence of the pheromone on the node selection.

4.3. Introduction of load balancing factor (LBF).

In the specific scheduling process, for the cloud computing resource pool is heterogeneous and dynamic, each virtual machine computing capacity, available bandwidth appear quite different due to the difference of geographical location. Some virtual machine performance may be better than other virtual machine. In this case, a large number of tasks can be allocated to a few of the high performance of virtual machine resources, the formation of waiting queue is long, and many virtual machines resources in idle state, which not only cause a huge waste of resources but also carries a non-satisfaction on the single task completion time. In order to solve this problem, we introduce load balancing factor(LBF) to each ant colony optimization round . It is recorded the load of each VM . The results will be used in the next round of pheromone update, increasing the load to the under load VM and reduce the pressure to the over load VM .Enhanced positive feedback effect.

$$LBF = 1 - (E_x - E_{min})/(E_{max} - E_{min}) \tag{8}$$

$$\tau_{ij}(t+1) = ((1-p)\tau_{ij}(t) + p\Delta\tau_{ij}(t)) * LBF \tag{9}$$

$$\tau_{ij}(t+1) = \tau_{ij}(t) * LBF \tag{10}$$

In this algorithm, the proposed algorithm is used to update the local pheromone and the global pheromone , By using the formula (9) and (10) to readjust the pheromone, the load balance factor is re adjusted by formula (8), E_x is the running time of Vm_x in the last iteration, E_{min} is the shortest running time of all virtual machines in the last iteration, and E_{max} is the longest running time of all virtual machines in the last iterations. If selected the specific virtual machine appears on the solution last round, use formula (9) readjust the pheromone, else use formula (10) readjust the pheromone , After the introduction of the LBF, we found VM heavy load in the iterative process of pheromone will be substantially reduced, light VM load in the iterative process of pheromone will rise sharply. By using subsequent roulette gambling algorithm, the VM with low load will be preferred to be chosen. Load balancing degree increases significantly.

4.4. **Pheromone update mechanism.**

$$\tau_{ij}(t+1) = ((1-p)\tau_{ij}(t) + p\Delta\tau_{ij}(t)) \tag{11}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{12}$$

In this paper, the M.Drigo model proposed by Ant-Cycle is used, and the P in the above formula of $1-p$, p is a volatile factor, $\Delta\tau_{ij}(t)$ is the pheromone increment of the path of $task_i$ to vm_j in t round. The increment is 0 when starts, $\sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$ is the total amount of pheromone that kth ant left in the path . In order to speed up the update of pheromone, the update is divided into three levels:

(1) local pheromone update

In an iterative process, regardless of the advantages and disadvantages of the ant's current round path, that the main work is to update the information on the path through the path. In the formula (13), $TotalTime_{ij}$ shows that ith task in VMj from submission to the completions time it takes(if there are tasks in front of it, including waiting time)

$$\Delta\tau_{ij}(t) = \frac{1}{TotalTime_{ij}} \tag{13}$$

(2) global pheromone update

The $TotalTime_{ij}$ in (13) is the spending time that for the current round of iteration in the optimal path to complete the task on the VMj.

(3) positive feedback

The global pheromone update (13) is used as the enhancement of positive feedback. With the iterations number increases, the positive feedback is gradually weakened, and the random of the search is enhanced, and the convergence rate is slowed down. Therefore, we use the optimal solution in the local and global update. In order to ensure that the update rate of pheromone, in order to highlight the difference between the quality of the path and the general path, to do a global positive feedback.

5. Experiment and analysis.

5.1. Experimental procedure and selection of parameters. From formula (7), the greater the value of the a, the greater the probability of choosing the local optimal path at a certain point. So the algorithm can improve the convergence speed of searching, but the problem is easy to fall into local optimal solution. The greater the value of pheromone factor b, the greater the probability that the path will be traversed by ants in a certain point, so that the performance of the algorithm is reduced if b×s value is too large. In summary, a factor determines the intensity of the deterministic and the prior factors in the process of the search path.And the b factor determines the intensity of the random factors in the process of the search path. We know the global optimization ability of an algorithm requires a strong randomness, but the convergence of the algorithm requires a strong determination. Therefore, the performance of the ant colony algorithm can be determined by the two interaction factors.

In order to verify the effectiveness and feasibility of this system, the simulation experiment is carried out by Cloudsim3.0. Datacenter, Cloudlet, Vm, Databroker, and some helper classes are extended and simulated to create cloud task and are used to submit it. Compared with the FCFS, the standard ant colony algorithm and the "cloud computing task scheduling algorithm based on load balancing ant colony optimization algorithm".

In the simulation platform, there are 3 resource centers, each resource center has 10 VM, the performance parameters of the virtual machine for $250-2000$mips range, roughly 4 levels, Heterogeneous environment of simulation resources, bandwidth $3000-10000b/s$, must perform $1000-6000$ cloud tasks. The length of the task instruction is $5000-15000$MI, which supports the space sharing strategy, That is, the Cloudlet Scheduler Shared strategy, the time to share; the default does not support virtual machine migration strategy. In the experiment, the 4 scheduling algorithms are compared with the number of iterations and different tasks. The convergence speed of the algorithm is tested. The performance of the algorithm is evaluated by the performance of the whole system. Ant colony algorithm iteration number is 30, a total of 8 ants participated in the search process.

6. experimental results and analysis. In the figure 2, a comparison of our algorithm (LBF-ant colony algorithm), HBB-LB[6], FCFS improved ant colony algorithm[7] and original ant colony algorithm in different conditions is given.

$$DI = \frac{E_{max} - E_{min}}{E_{avg}} \tag{14}$$

In the formula (14), DI represents degree of imbalance, E_{max} represents longest running time the last round in all virtual machines, E_{min} represents shortest running time the last round in all virtual machines, E_{avg} represents average running time the last round in all virtual machines.

We can see from it that the algorithm proposed by this system is significantly improved compared with the original ant colony algorithm. HBB-LB and the improved ant colony algorithm in the total completion time. With the increase of the amount of the task, the ant colony is gradually falling into the load of imbalance, which leads to the decrease of resource utilization rate and the time of completion time is longer. The load balance factor introduced by the system and the dynamic adjustment of the pheromone volatile factor can be very good in the specified search times to find a satisfactory solution.As for the improved ant colony algorithm,with the increase of the amount of tasks,it is premature

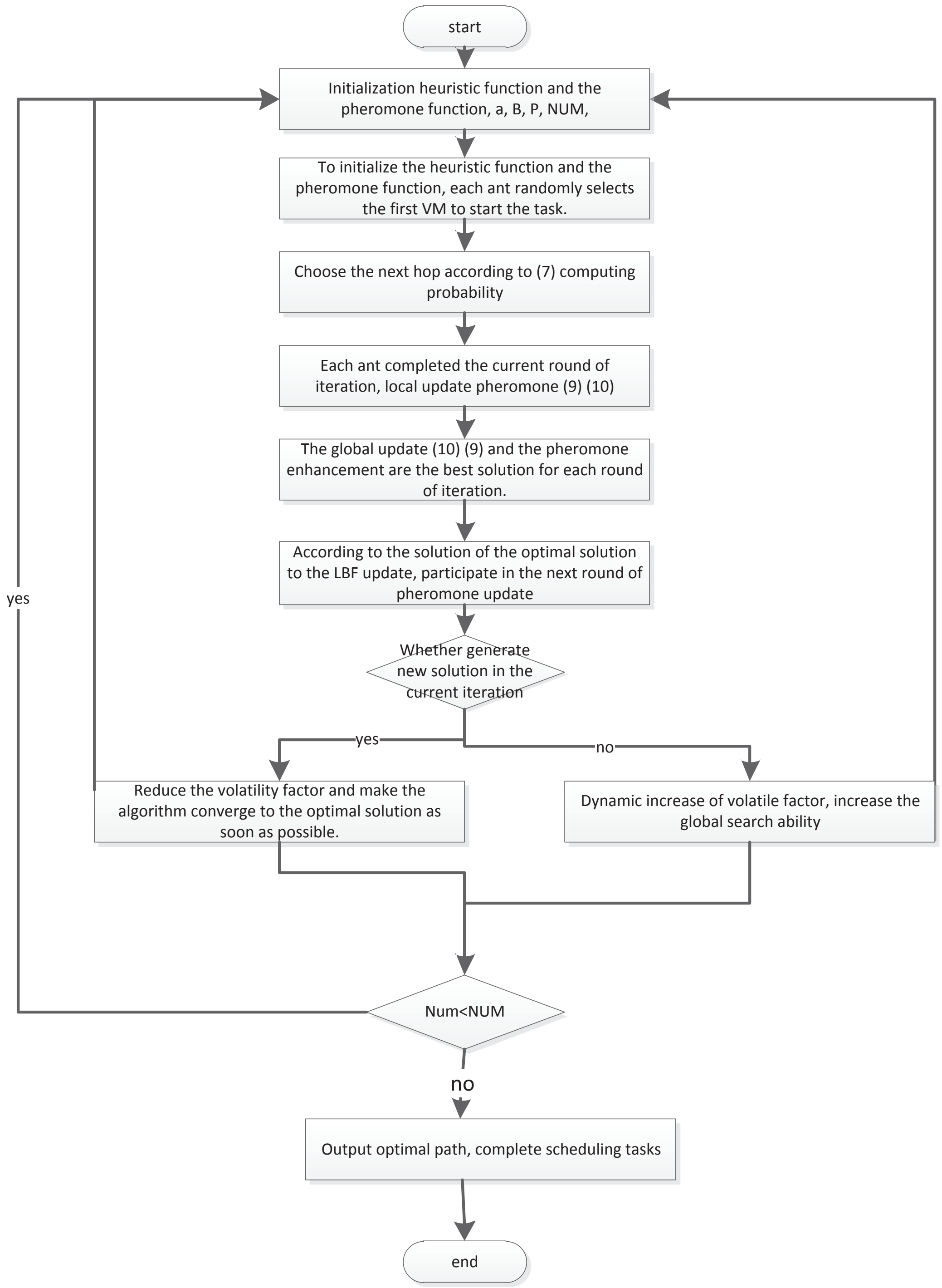

Figure 1. The flow chart of the scheduling algorithm

convergence to local optimal solution.It is further proved that our algorithm has a better tradeoff between certainty and randomness.

From figure 3 we can see that in the homogeneous cluster environment, the performance parameters of the virtual machine is about 2000mips, the bandwidth capacity is 10000b/s,

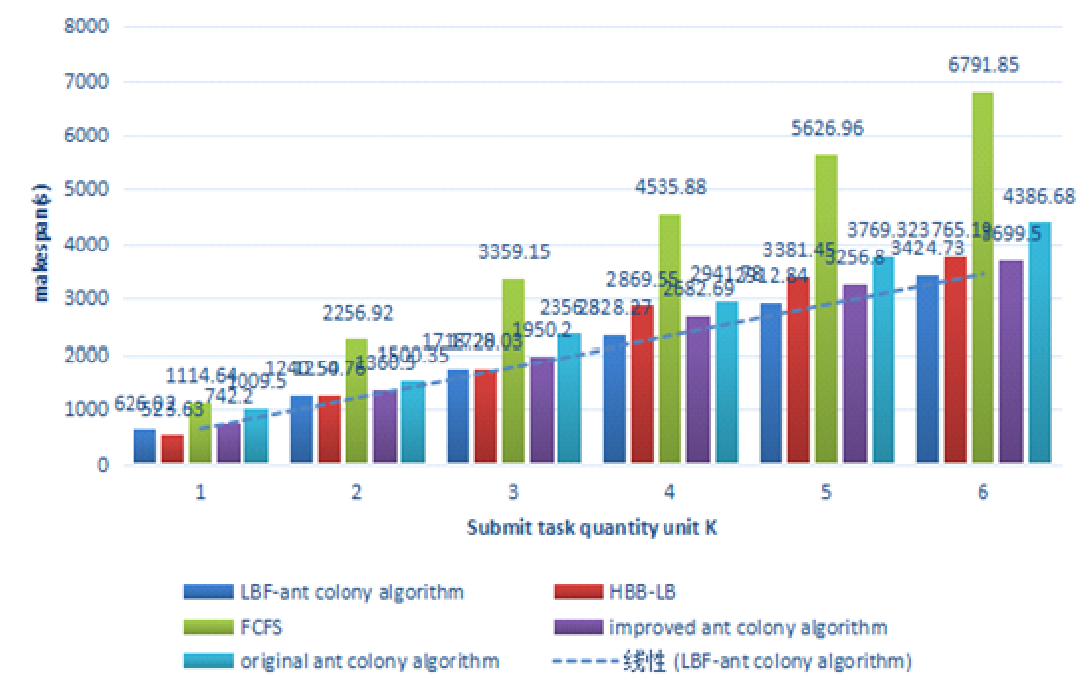

FIGURE 2. Comparison of the results of the scheduling algorithm

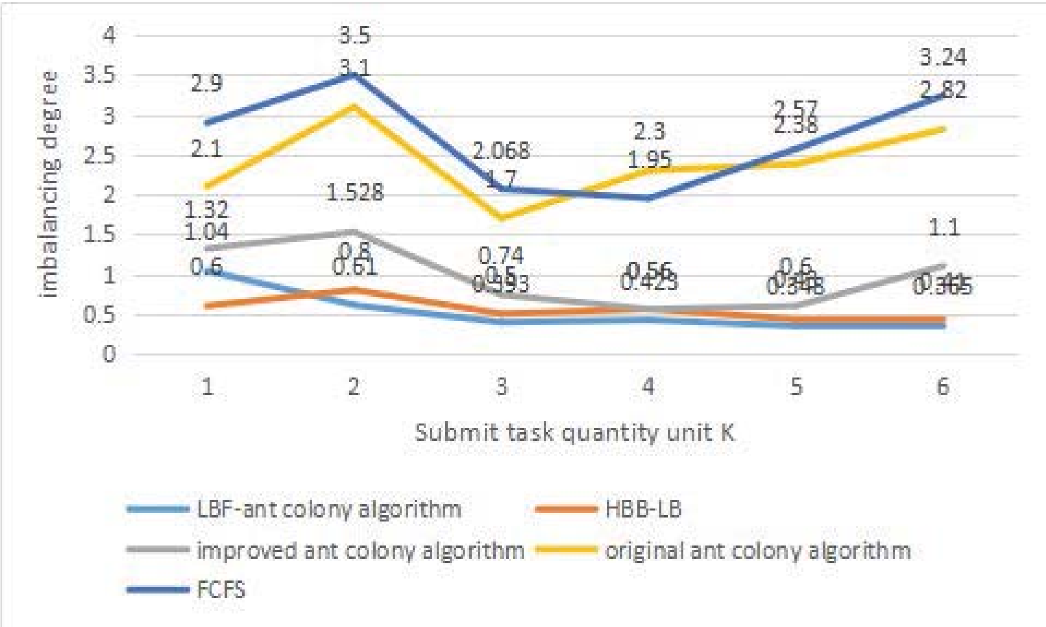

FIGURE 3. Comparison of unequal results based on the non-degree (Comparison of different results in homogeneous environment)

the number of tasks is from 1000 – 6000, the algorithm proposed by this system is more stable at around 0.4, and the overall system is more balanced.

From figure 4 we can see that in heterogeneous environment, The heterogeneous environment is simulated by the following conditions:the performance parameters of the virtual machine are divided into 4 grades, range 250 from 2000 mips.bandwidth performance is also different.the simulation due to different factors such as geographical location.

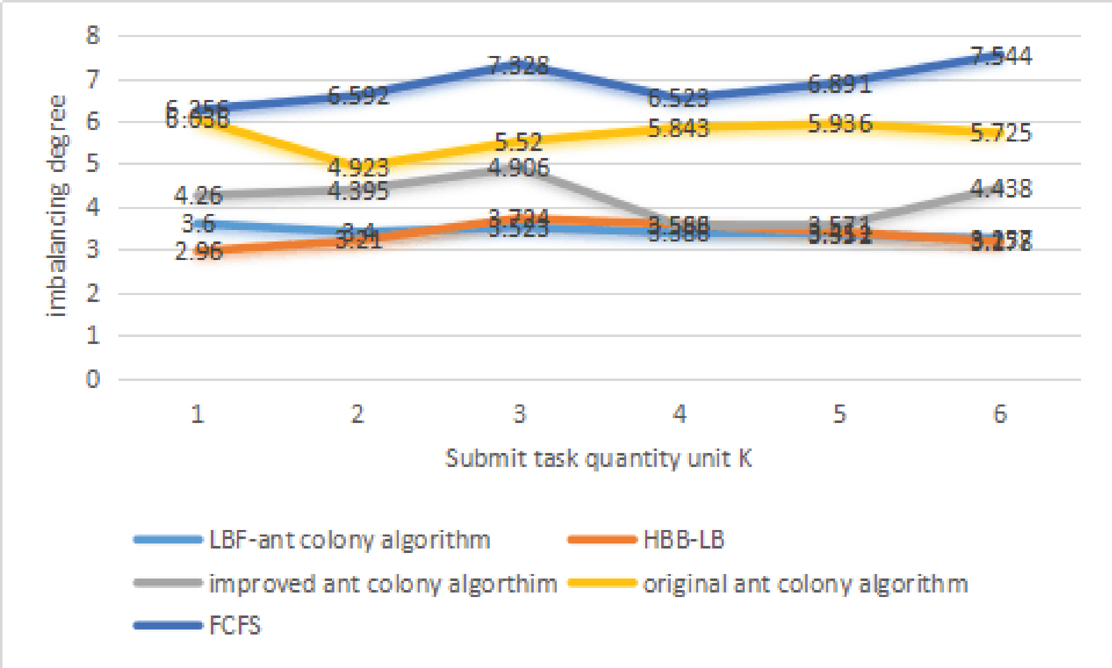

FIGURE 4. Comparison of unequal results based on the non - degree (Comparison of different results in heterogeneous environment)

From the graph we know that the algorithm in heterogeneous environment has the more obvious effect, Compared with the HBB-LB algorithm, the load is almost the same. DI of FCFS and the original ACO is generally higher than the average of the first two,the reason is due to the lack of load balancing factor adjustment function.so that the VM which has good performance may handle $200 - 300$ tasks, the bad may handle $10 - 60$ tasks , resulting in a serious decline in the performance of the whole system.

7. **Conclusions.** Generally speaking,our proposed algorithm, the LBF factor is introduced in our system.And the idea of local and global information is introduced in the process of updating pheromone. And the tasks waiting time is used as a penalty factor to accelerate the flow of pheromone. Each round of scheduling add the adjustment of the load balance factor in the last round. At last, according to the actual situation of the ant colony optimization, adjusting to increase or decrease the global search ability. Experimental results show that the proposed algorithm can effectively reduce the task completion time and also have an effective control on the system with the degree of inequality. Especially in the network, computing performance of a serious heterogeneous state, the system of the uneven degree of comparison of other algorithms can also be in a good state. The next step,we will focus on the study of flow task scheduling model and user priority scheduling model, so that make the system is more realistic, more lean close to realistic demand.

Acknowledgment. The authors would like to thank for the support from the project Innovation Fund of Harbin Institute of Technology with the Grant number:
HIT. NSRIF.2015087, the Shenzhen science and technology basic research project with the Grant number: JCYJ20150403161923527 and the Shenzhen science and technology techniques research project with Grant number: JSGG20150330103937411.

REFERENCES

[1] R. S. Chang, C. Y. Lin, and C. F. Lin, An Adaptive Scoring Job Scheduling algorithm for grid computing. *Information Sciences Elsevier*, vol.207, no. 1, pp. 79-89, 2012.

[2] F. Farahnakian, T. Pahikkala, and H. Tenhunen, Using Ant Colony System to Consolidate VMs for Green Cloud Computing. *IEEE transactions on services computing*, vol. 8, no. 2, pp. 188-196, 2015.

[3] J. Liu, G. Li, An Improved MIN-MIN Grid Tasks Scheduling Algorithm Based on QoS Constraints. *Proc. of International Conference on Optics Photonics and Energy Engineering (OPEE)*, vol. 1, pp. 281 - 283, 2010.

[4] D. H. Miriam, K.S. Easwarakumar, A Double Min Min Algorithm For Task Metascheduler on Hypercubic P2P Grid Systems. *International Journal of Computer Science Issues*, vol.7, no.4, pp. 8-18, 2010.

[5] H. K. Chen, F. Wang, and N. Helian, User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing. *International Journal of Computer Applications IJCA*, vol. 9, no. 4, pp. 24-31, 2014.

[6] D. Babu, V. Krishna, Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, vol. 13, pp. 2292C2303, 2013.

[7] Braun, Tracy, DA comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing*, vol. 61, no. 6, pp. 810 C 837, 2001.

[8] B. Chang, H. Tsai, C. F. Huang, Z. Y. Lin, C. M. Chen, Fast access security on cloud computing: ubuntu enterprise server and cloud with face and fingerprint identification. *Proc. of the 2nd International Congress on Computer Applications and Computational Science*, pp.451C457, 2012.

[9] A. M. Bernardino, E. M. Bernardino, J. M. Snchez-Prez, M. A. Vega-Rodrguez,J.A. Gmez-Pulido, Efficient Load Balancing Using the Bees Algorithm. *Trends in Applied Intelligent Systems, Lecture Notes in Computer Science*, 2011.

[10] Q. K. Pan, M. F. Tasgetiren, P. Suganthan, T. Chua. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, vol.181, no. 12, pp.2455C2468, 2011.

[11] M. Aruna, D. Bhanu, P. Suganthan, R.Punithagowri. A Survey on Load Balancing Algorithms in Cloud Environment. *International Journal of Computer Applications*, vol.82, no.16, pp. 39-43, 2013.

[12] J. S. Pan, P. F. Ren, L. L. Tang. Research on Heuristic Based Load Balancing Algorithms in Cloud Computing.*Advances in Intelligent System and Computing*, vol. 370, pp. 417-426, 2015.