*Journal of Information Hiding and Multimedia Signal Processing* ©2018 ISSN 2073-4212
**Ubiquitous International** Volume **9**, Number **6**, November **2018**

# A New Hybrid Task Scheduling Algorithm Designed Based on ACO and GA

Linlin Tang[1], Xi Zhang[3], Zuohua Li[1] and Yong Zhang[2]

[1]Harbin Institute of Technology, Shenzhen

[2]College of Information Engineering, Shenzhen University

[3]Department of Basic Teaching, Shenzhen Technology University
zhangxi@sztu.edu.cn

Received July, 2018; Revised August, 2018

Abstract. *As known to us all, task scheduling is one of the most important problems in Cloud Computing. For this, a new hybrid algorithm named GAPACO which considers both cost-saving and time-saving has been proposed in this paper. Especially, it combines two advantages of ACO and GA. It takes advantage of rapid convergence of ACO and simpleness of GA. A large number of experiments data shows that this hybrid algorithm is available and can be applied at community cloud environment. The experimental results show its efficiency.*
**Keywords:** Cloud Computing, Task Scheduling, Hybrid Algorithm, Cost Saving

1. **Introduction.** With rapid development of the Internet, Cloud Computing[1] has been paid more and more attention as a new technique trend. It focuses on delivering software, platform and infrastructure to the users by cloud service provider. Cloud Computing is experiencing rapid development both in academy area and in industry area[2], it is tremendously promoted by the business rather than academic. Cloud Computing as a kind of computer paradigm[3] has made computing resources as utility another step closer to the reality[4]. Following Figure 1 gives a typical Cloud Computing environment involves Cloud Service, Cloud Platform, Cloud Infrastructure and Cloud Storage. However, in this field, there are still many problems need to be solved. One of them is the well-known task scheduling problem. It is an NP-hard problem.

As we know that swarm intelligence algorithms[24, 25, 26, 27] are efficient ways to solve the combinatorial optimization problems. Single swarm Intelligence algorithms always focus on the time-saving[5, 6], they do not take cost into consideration. As alternative, many hybrid swarm intelligence algorithms proposed are intended to correct some defects in those single intelligent algorithms and to make use of their advantages at the same time. Some researchers pay attention to load balance [7, 8] or other aspects[9]. For example, the Ant Colony algorithm[10] may ensure to get the global optimal solution. In fact, it is so time-consuming that it cannot be appropriately applied onto some small clouds such as private cloud or community cloud.

Many papers have proved that any optimization algorithm for scheduling problem can get their ideal result regardless of time and space complexity. But there are so many restraints and limits we have to pay attention. A new hybrid algorithm called GAPACO which can meet general evaluation criterions, especially time-saving and cost-saving meanwhile is proposed in this paper. Some experiments are used for testing. The experiment

results indicate that this algorithm is more fit for the community cloud environment compared with the ACO algorithm[8]. It can save users' cost with fewer total time loss at the same time.

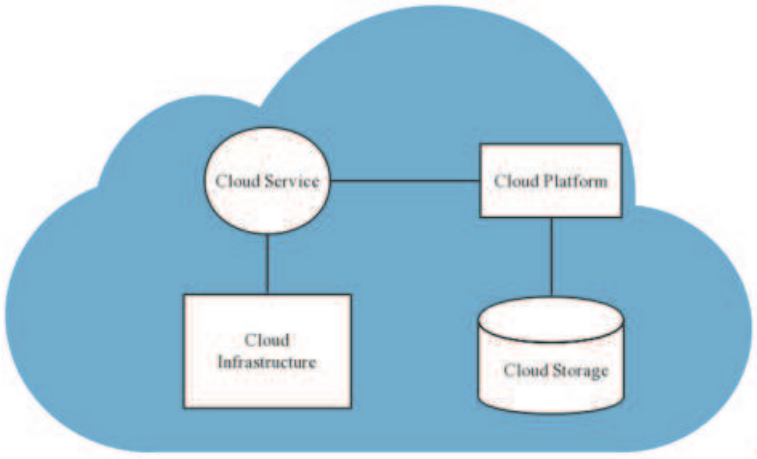


FIGURE 1. Typical Cloud Computing Architecture

The remainder of the paper is organized as follows: In Section 2, some related work is introduced. In Section 3, the GAPACO algorithm is proposed. In Section 4, experiments are performed. Based on experiment results, a conclusion and discussion for future research is given in Section 5.

## 2. Related Work.

**2.1. Classical Algorithms Applied in the Early Stage.** Many researchers apply scheduling policies onto task scheduling problems. For example, O.M.Elzeki[11] proposed an improved Max-min algorithm. This algorithm achieves simplicity and efficiency. Both Max-Min and Min-Min[12] all use one simple strategy. Ajay Gulati[13] proposed a dynamic round robin for load balancing in cloud computing. Zhou Zhou[14] proposed an algorithm based on greedy algorithm. Gulshan Soni[15] proposed a novel approach for load balancing in cloud data center. They used priority based on request at the central load balancer. Atul Vikas Lakra[7] proposed a multi-objective tasks scheduling algorithm for throughput optimization. It selects a suitable virtual machine based on query, then assigns QoS for tasks and VMs.

**2.2. Individual or Hybrid Intelligence Algorithms.** Furthermore, some other researchers use various kinds of swarm intelligence algorithms based on individual population. Kousik Dasgupta[16] proposed a genetic algorithm based load balancing strategy. Hongnan Zhao[6] proposed a kind of interaction artificial bee colony based load balance method. This algorithm gives a new computing method of load balance. A.I.Awad[17] proposed an enhanced particle swarm optimization for task scheduling. It firstly gives a cloud structure mainly including scheduler and task and resource information collector. Then they proposed LBMPSO algorithm to fit this model.

$$bla = \sqrt{\sum (\frac{\sum p_i}{vmNum} - P_i)^2 / vmNum} \tag{1}$$

Some researchers give two or more kind of swarm intelligence algorithms to be combined into one for advantages. For example, Dr.M.Sridhar[18] proposed a hybrid genetic swarm scheduling for cloud computing. He combined the Particle Swarm Optimization (PSO) with the Genetic Algorithm (GA) named GAPSO. It performs better when compared with the Max-Min scheduling in execution time. Yi-Tung Kao[19] proposed another hybrid genetic algorithm and particle swarm optimzation for multimodal functions. It sequentially uses GA and PSO. This algorithm is testified simple and effective to handle different kinds of continuous optimization problems.

## 3. **Our Proposed Method.**

3.1. **Introducion of CloudSim Environment.** To measure efficiency and effecitveness of hybrid algorithm, some experiments are needed. CloudSim toolkit[21] has been chosen as simulation platform. CloudSim[22] is built in CLOUDS(Cloud Computing and Distributed System) Laboratory by the University of Melbourne, Australia. The following Figure 2 shows the working process of CloudSim[9].

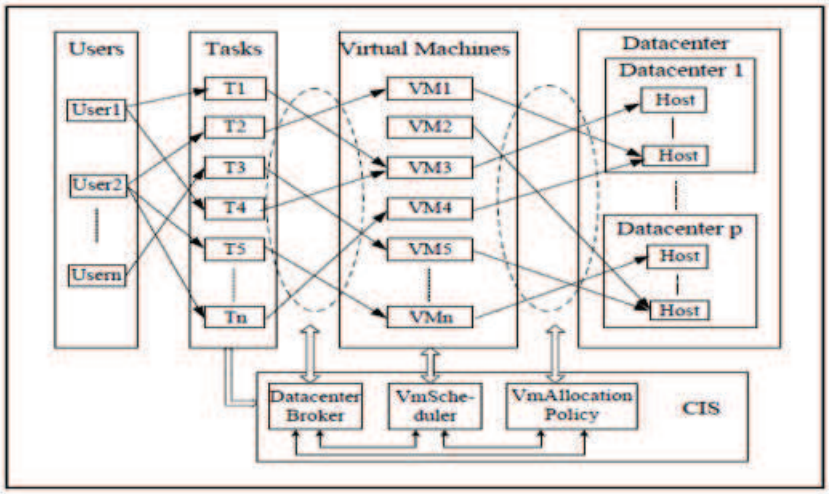


FIGURE 2. CloudSim Working Process

From this figure,we can give a clear description of task scheduling problem. In general, the tasks from different users are relatively independent. We consider n independent tasks as $T_1, T_2, T_3, \cdots$ and n virtual machines as $VM_1, VM_2, VM_3, \cdots$. Our target is to assign these n tasks to the n virtual machines according to some distribution requirement.

3.2. **Mathematical Model.** Ant colony algorithm derives from Travelling Salesman Problem[23]. However cloud task scheduling is abnormal TSP. We assign N tasks to M virtual machines without any task left. Figure 3 shows this process.

The total number of iterations of ant colony algorithm is MAX_GEN. When the previous iteration is completed, pheromone matrix released by every ant is $\Delta\tau$.

$$\Delta\tau = \begin{bmatrix} \Delta\tau_{00} & \cdots & \Delta\tau_{0m} \\ \vdots & \ddots & \vdots \\ \Delta\tau_{n0} & \cdots & \Delta\tau_{nm} \end{bmatrix} \tag{2}$$

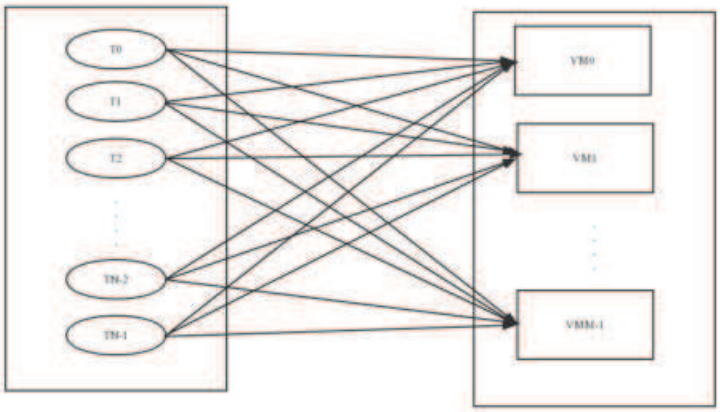

FIGURE 3. ACO Applied to Task Assignment Problem

The idea of ACO algorithm is that every ant finds path in each iteration. Firstly, iteration is initialized by random method. When the previous iteration is finished, pheromone matrix of every ant updates the global pheromone matrix. The updated pheromone matrix guides path is used to find the next generation. Pheromone is global shared pheromone matrix. This kind of matrix indirectly indicates which path segment is the shortest. rho is a weight parameter. Pheromone updating value is reciprocal of the M virtual machine running time. Assume that $ant_i$ finds a path $seqExecute$, every virtual machine time-consuming in this path is $costVm[1 \cdots M]$. So ant's pheromone updating value named Delta can be demonstrated by the following formula (4). $Q$ is a constant parameter. Pheromone matrix plays a conclusive role in finding the next (task, vm) two-tuples. It is showed in formula (5). Here, $P_{ij}^k(t)$ denotes probability of ant $k$ is assigned $T_i$ to $VM_j$ at $t$ moment. After that, swarm intelligence algorithm results in the pheromone on path segment which help to shorten time of all tasks.

$$pheromone \begin{bmatrix} p_{00} & \cdots & p_{om} \\ \vdots & \ddots & \vdots \\ p_{no} & \cdots & p_{nm} \end{bmatrix} = pheromone \begin{bmatrix} p_{00} & \cdots & p_{om} \\ \vdots & \ddots & \vdots \\ p_{no} & \cdots & p_{nm} \end{bmatrix} * *(1 - rho) \tag{3}$$

$$Delta = Q/max(costVm) \tag{4}$$

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [r_{ij}(t)]^\beta}{\sum_{l \in AllowedTask(t), h \in vms} [\tau_{lh}(t)]^\alpha [r_{lh}(t)]^\beta} \\ 0, otherwise \end{cases} \tag{5}$$

Hybrid swarm intelligence algorithm GAPACO combines both advantages of GA and ACO and avoid their disadvantages. This algorithm makes full use of the GA to boost ant colony algorithm. It turns out to expand the superior of ACO as far as possible, at the same time to rectify their disadvantages. Hybrid algorithm excels to both single algorithms in various performance indexes in the end. By modifying the fitness function of GA, it makes the hybrid algorithm to go towards time-saving and cost-saving orientation. The biggest problem of ACO is that running time is huge in ACO on every generation. So we control the iteration times to be 2 at first stage of hybrid algorithm. On the contrary, the cost of every generation of GA is so short that we should relax the control of iteration times. GA selects the chromosome from the previous generation population based on

fitness greater than average fitness value. So after some iteration times, fitness value will be gradually improved. Formulas (6) and (7) show the key innovation point of this part.

$$Fitness = Q'/(r * max(costVm) + (1 - r) * Debt) \tag{6}$$

$$\begin{aligned} Debt = {} & peNmu * costPe * \sum_{i \in vms} costVm(i) \\ & + costPerMem * ram + costPerStorage * size + costPerBw * bw \end{aligned} \tag{7}$$

In (6), $r$ control weight between speed of completion and spending of cost. $Q'$, is parameter used for common magnification. In (7), peNum is the number of pe in a single virtual machine.costPe is the cost of single pe. costPerMem is cost of memory. ram is capacity of memory. costPerStorage is the cost of storage. size is the size of storage. costPerBw is the cost of bandwidth. bw is the amount of bandwidth.

The idea of hybrid swarm intelligence algorithm is that it is firstly to run the ACO algorithm when the hybrid algorithm gets an intermediate result, then compute their fitness value. After some generation iteration, hybrid algorithm goes towards minimizing the user's cost and load balance. From table 2, we can see that ACO algorithm just needs a few not so many iteration times and it can get a acceptable solution. Adding iteration times of ACO can not help find better solution or improve current solution, and this will also course a tremendously increase of the running time. Particularly, pheromone matrix at the end of previous generation can be continued to be used for the next generation. In our hybrid algorithm, the iteration time of ACO is set to 2, and the iteration time of GA may be greater than 2, with considering GA not very time-consuming. Some experiments have been applied on this hybrid algorithm. The results show that hybrid algorithm can save time and money as much as possible.

4. **Experimental Results and Analysis.** Two groups experiments are performed here, one is based on the self-control experiment, another is based on the blank control experiment against Medhat A. Tawfeeks paper[8]. Results are shown in the following table 1. For simplicity, we assume the Total number of task and the Total number of VMs are constants.

In order to get accurate data, two observation points are introduced. For Table 1, Gen_GA is the number of generation of GA stage part. We control that the generation number of the first stage is 2. Money-saving whose unit is CNY meas difference of solutions of two observation points. ACO/GA makespan time whose unit is second means makespan of completing the tasks. ACO/GA assigning time whose unit is second means the time-cost of finding the path of assigning all task to all available virtual machines. The total running time is the sum of makespan time and assigning time.

Table 1 is the self-control experiment. Iteration time of first stage is 4, the number of task is 600 and the number of virtual machine is 30. Then we gradually change the iteration time of second stage. At last, we get the figures above. From them, we can see that the more iteration time, the more cost-saving. And we can find that ACO Makespan Time is approximately equal to the GA Makespan Time. But ACO Assign Time is far more than GA Assign Time. Considering time-saving, we should set that GEN_GA should less than 4.

In order to compare with Medhat A. Tawfeeks idea[8], some other experiments are introduced. For the total task number is constant, we adopt the offline algorithm in Medhat A. Tawfeeks paper. ACO algorithm in Medhat A. Tawfeeks paper is used as compared experiment. In Table 2, GEN_ACO is thegeneration number of ACO stage in

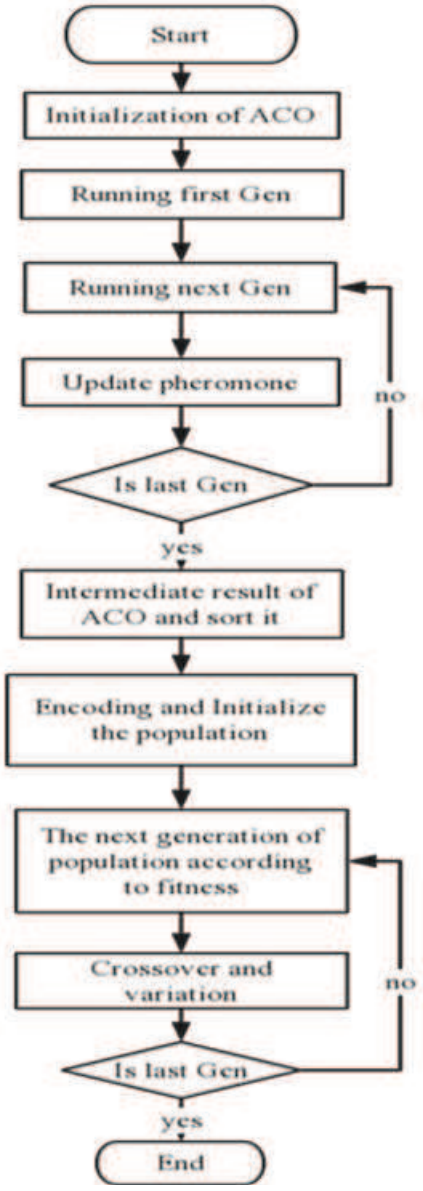


FIGURE 4. Flow Diagram of Our Proposed Algorithm

GAPACO. GEN_GA is the generation number of GA stage in GAPACO. Cost_hybrid whose unit is CNY is the total cost money of hybrid algorithm. Cost_ACO whose unit is CNY is the total cost of ACO algorithm in Medhat A. Tawfeeks paper [8]. Let the hybrid algorithm and ACO have the same total running time, then we get the following results.

From above, experimental results show that hybrid algorithm GAPACO can substantially performs better at saving users cost.

5. **Conclusions.** Based on the ACO algorithm, a hybrid algorithm aiming at meeting users demand has been proposed here. Experiments show that GAPACO can get a good results at saving users money. During this research, a new pheromone update policy is introduced called the Generational Local Policy(GLP). But, the hybrid algorithm as a

TABLE 1. Parameters Setting of CloudSim

| Item | Parameters | Value |
|---|---|---|
| Task(Cloudlet) | Length of Task | 300000-3600000 |
| | Total number of Task | 200 |
| | Inputsize | 10 |
| | Outputsize | 10 |
| Virtual Machine | Total Number of VMs | 10 |
| | MIPS | 500-1500 |
| | VM memory(RAM) | 128 |
| | Bandwidth | 100 |
| | Cloudlet Scheduler | Space_shared and Time_shared |
| | Number of Pes requirement | 2 |
| | Image size | 10000 |
| | VMM | Xen |
| Unit Price(cent) | costPe | 3.0 |
| | costPerMem | 0.05 |
| | costPerStorage | 0.003 |
| | costPerBw | 0.001 |

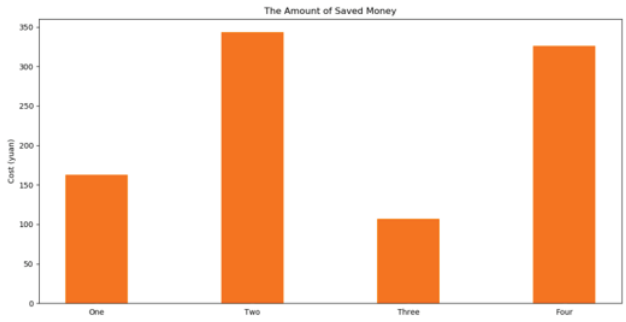


FIGURE 5. Saved Money and The Iteration of GA Part

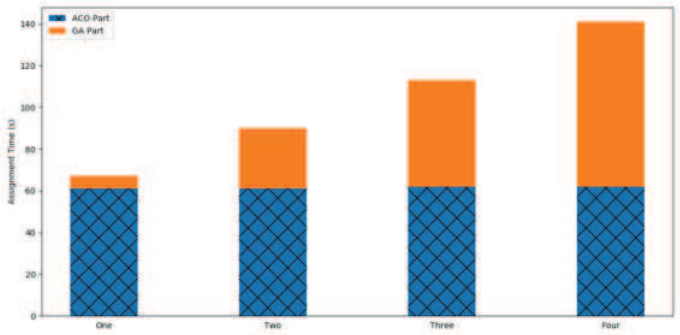


FIGURE 6. Assignment Time and The Iteration of GA Part

kind of heuristic algorithm can not stay more steadily than the single one algorithm. Our future research will emphasis on improving the stability.

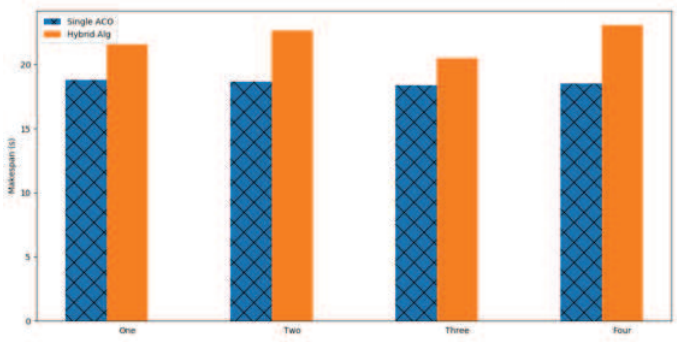


FIGURE 7. Makespan and The Iteration of GA Part

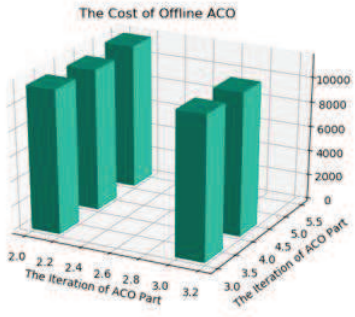


FIGURE 8. The Cost of Offline ACO Alg

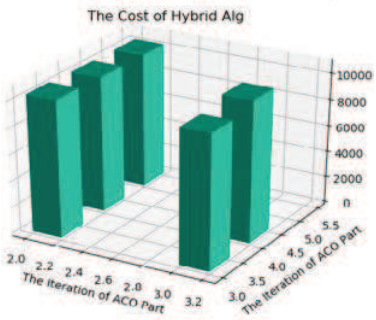


FIGURE 9. The Cost of Hybrid Algt

**Acknowledgment.** The work was supported in part by the Science and Technology Plan Projects of Shenzhen with grant number *JCYJ*20170302145623566. It is also partly supported by a grant from Shenzhen Technology University with grant number 2018010802008.

## REFERENCES

[1] S. Azodolmolky, P. Wieder, R. Yahyapour Cloud Computing Networking: Challenges and Opportunities for Innovations. *Communications Magazine, IEEE*, vol. 51, no. 7, pp.59-63, 2013.

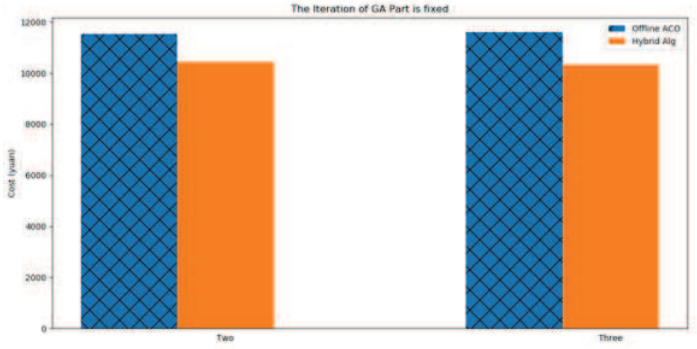


FIGURE 10. Cost and The Iteration of ACO Part

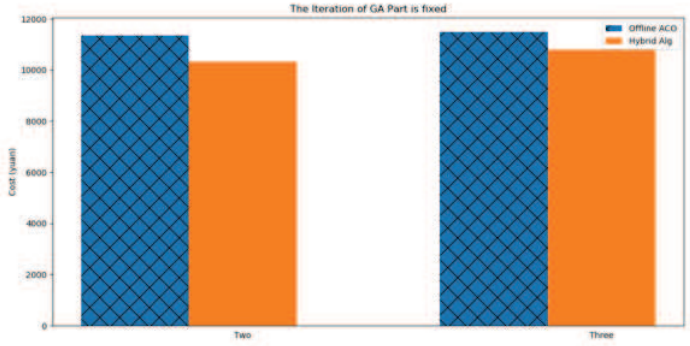


FIGURE 11. Cost and The Iteration of ACO Part

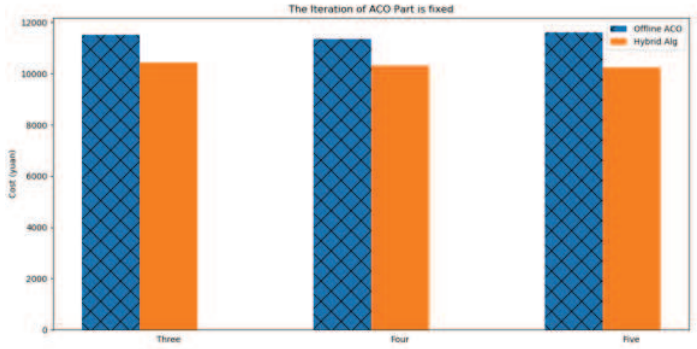


FIGURE 12. Cost and The Iteration of GA Part

[2] K. Li , G. Xu , G. Zhao , et al., Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. *Proc. of Chinagrid Conference (ChinaGrid), 2011 Sixth Annual. IEEE*, pp. 3-9, 2011.
[3] Cloud Computing: Principles and Paradigms. *Publisher: Wiley Press, New York, USA*, 2010.
[4] A. Beloglazov , Energy-efficient Management of Virtual Machines in Data Centers for Cloud Computing. *Department of Computing and Information System*, 2013.
[5] L. Guo , S. Zhao , S. Shen , et al., Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm. *JNW*, vol. 7, no. 3, pp.547-553, 2012.

[6] L. L. Tang, J. S. Pan, Y. Y. Hu, P. F. Ren, Yu Tian, Hongnan Zhao, A Novel Load Balance Algorithm for Cloud Computing. *Proc. of ICGEC2015*, vol. 388, pp. 21-30, August 2015.
[7] Y. Fang, F. Wang, J. Ge, A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. *Proc. of International Conference on Web Information Systems and Mining*. Springer Berlin Heidelberg, pp.271-277, 2010.
[8] J. S. Pan ,H. Wang , H. Zhao , et al., Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing. *Genetic and Evolutionary Computing*. Springer International Publishing, pp. 49-57, 2015.
[9] A V. Lakra, D. K. Yadav, Multi-objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. *Procedia Computer Science*, vol. 48, pp.107-113, 2015.
[10] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, et al., Cloud Task Scheduling Based on Ant Colony Optimization. *Proc. of Computer Engineering and Systems (ICCES), 2013 8th International Conference on. IEEE*, pp.64-69, 2013.
[11] O. M. Elzeki , M. Z, Reshad, M. A. Elsoud , Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*, vol. 50, no. 12, pp.22-27, 2013.
[12] M. Y. Wu, W. Shu , H. Zhang , Segmented Min-Min: A Static Mapping Algorithm for Meta-tasks on Heterogeneous Computing Systems. *Proc. of Heterogeneous Computing Workshop, 9th. IEEE*, pp.375-385, 2000.
[13] A. Gulati , R. K.Chopra , Dynamic Round Robin for Load Balancing in a Cloud Computing. *IJCSMC*, vol. 2, no. 6, PP.274-278, 2013.
[14] L. Ma , Y. Lu , F. Zhang , et al., Dynamic Task Scheduling in Cloud Computing Based on Greedy Strategy. *Proc. of International Conference on Trustworthy Computing and Services*. Springer Berlin Heidelberg, pp.156-162, 2012.
[15] G. Soni , M.. Kalra , A Novel Approach for Load Balancing in Cloud Data Center. *Proc. of Advance Computing Conference (IACC), 2014 IEEE International. IEEE*, pp.807-812, 2014.
[16] K. Dasgupta ,B. Mandal ,P. Dutta , et al., A Genetic Algorithm (GA) Based Load Balancing Strategy for Cloud Computing. *Procedia Technology*, vol. 10, pp.340-347, 2013.
[17] A. I. Awad, N. A. El-Hefnawy , H. M. Abdel-kader , Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments. *Procedia Computer Science*, vol. 65, pp.920-929, 2015.
[18] M. Sridhar, Hybrid Genetic Swarm Scheduling for Cloud Computing. *Global Journal of Computer Science and Technology*, vol. 15, no. 3, pp.9-16, 2015.
[19] Y. T. Kao, E. Zahara , A Hybrid Genetic Algorithm and Particle Swarm Optimization for Multimodal Functions. *Applied Soft Computing*, vol. 8, no. 2, pp.849-857, 2008.
[20] M. Katyal, A. Mishra , A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. *Computer Science*, vol. 6, no. 1, pp.25-36, 2014.
[21] R. N. Calheiros, R. Ranjan, A. Beloglazov , et al., CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Software: Practice and experience*, vol. 41, no. 1, pp.23-50, 2011.
[22] CloudSim. (2016, November 14). *In Wikipedia, The Free Encyclopedia*. Retrieved 07:24, April 26, 2017
[23] Ant colony optimization algorithms. (2017, March 26). *In Wikipedia, The Free Encyclopedia*. Retrieved 07:28, April 26, 2017
[24] S. C. Chu, J. F. Roddick and J. S. Pan, Ant Colony System with Communication Strategies. *Information Sciences*, vol. 167, no. (1-4), pp.63-76, 2004.
[25] S. C. Chu, J. F. Roddick, C. J. Su and J. S. Pan, Constrained Ant Colony Optimization for Data Clustering. *Proc. of 8th Pacific Rim International Conference on Artificial Intelligence*, LNAI 3157, pp. 534-543, 2004.
[26] M. Zhao, J. S. Pan, C. W. Lin, L. J. Yan, Quantification-Based Ant Colony System for TSP. *Proc. of ICGEC 2013*, pp.331-339, 2013.
[27] Y. T. Chen, M. F. Horng, C. C. Lo, J. S. Pan, S. C. Chu, A New Scheme of Ant Colony System Algorithm to Discovery Optimal Solution with Flip-flop Search. *Proc. of SMC 2011*, pp.925-930, 2011.