ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20220712164019
Original scientific paper

# An Improved Ant Colony Algorithm for New energy Industry Resource Allocation in Cloud Environment

Haoyang DU, Junhui CHEN*

**Abstract:** The new energy industry development is affected by many factors. Among them, the resources utilization ratio is a major reason for the low productivity of enterprises. As the core problem of cloud computing, the resource allocation problem has been widely concerned by the people, and the resource allocation problem of the new energy industry as the key to energy innovation and transformation should be more paid attention to. In multi-resource cloud computing scenarios, requests made by users often involve multiple types of resources. Traditional resource allocation algorithms have a single optimization object, typically time efficiency. In order to achieve cluster load balancing, utilization of system resources and improvement of system work efficiency, this paper proposes a new cloud computing allocation algorithm based on improved ant colony algorithm. According to the limit conditions of cloud computing environment and computing resources, this paper finds the shortest response time of all resource nodes and gets a set of best available nodes. This method can meet the quality requirements of cloud computing, and the task completion time of the improved algorithm is shorter, the number of algorithm iterations is less, and the load balancing effect is better. Through MATLAB simulation experiments, the effectiveness of the proposed method is verified.

**Keywords:** cloud computing; improved ant colony optimization; new energy industry; resource allocation

## 1 INTRODUCTION

In the context of global energy cooperation and energy structure transformation, with the continuous development of science and technology, the data of the new energy industry grows in a massive way, which arises at the historic moment in order to cope with the massive data processing and cloud computing. In order to efficiently handle multiple tasks proposed by multiple users in the cloud computing scenario, efficient resource allocation algorithms are essential [1, 2]. A core problem in cloud computing is load balancing, and Load balancing is defined as assigning tasks equally to individual resource point in a cloud computing system. So it is very important to design an efficient and reasonable resource allocation algorithm. Excellent resource allocation algorithm can provide multiple users to handle computing tasks faster under the same resource conditions, provide a good use experience, and have a high economic value.

Cloud computing refers to the supercomputing model connected through the Internet, including related technologies of distributed processing, parallel processing, Web2.0 and grid computing, or commercial implementation of these computer science concepts. Cloud computing is a new type of shared infrastructure that connects huge pools of systems to provide users with a variety of storage and computing resources over the Internet in the form of carriers and customers [3]. Computational resource allocation is an important component of cloud computing technology. Its efficiency directly affects the performance of the entire cloud computing environment [4, 5]. Due to the unique nature of cloud computing, the original resource allocation and scheduling algorithms for grid computing are no longer available. Working effectively in this environment, the improved Ant colony optimization allocation algorithm proposed in this paper comprehensively considers a series of characteristics of cloud computing, in order to efficiently allocate appropriate computing resources for user operations in this environment.

## 2 PROBLEM DESCRIPTION

Cloud computing evolves from grid computing and uses grid computing as its backbone and basic structure. It can be said that cloud computing is a more advanced form of grid computing. But there is a huge gap between the two in reality. The difference can be found in the literature [6]. Cloud computing provides more abstract resources and services. These resources and services can be divided into three levels, namely Software as a Service and Platform as a Service (Platform as a Service) and Facilities as a Service [7, 8].

One of the operational goals of the cloud computing environment is to use computing power as a service for users to purchase. Enterprises and individual users no longer need to invest in expensive hardware acquisition costs, only need to buy or lease computing power through the Internet, take their own computer as the entrance; everything is connected to the cloud computing environment through the Internet, and the acquisition of computing power is completed. However, this also indicates that the scale of cloud computing will be very large, and the exclusive resource allocation mode different from the grid, the entire cloud domain. The resources in the middle will be shared by all users at the same time to ensure that the delay-sensitive jobs will run well on the cloud. This means that the user's jobs in the cloud will be divided into processes or even thread-level levels. Therefore, cloud computing has stringent requirements for computing resource allocation algorithms. The computational resource allocation algorithm that this paper focuses on is to maximize the efficiency of the cloud computing environment by allocating the most appropriate computing resources to storage nodes.

## 3 RESOURCE ALLOCATION METHOD

### 3.1 Computing Resource Allocation Process

Referring to the cloud computing framework proposed by Map/Reduce [8, 9], each cell in the cloud environment consists of a single master job scheduling node and a slave task allocation node in each node cluster under the control of the cell. The master node is responsible for scheduling all tasks that constitute a job, and the data resources of these tasks are distributed in different slave sections [10]. In the user image fragmentation of point storage resources, the master node monitors their execution, re-executes failed tasks, or does error handling. After the slave node is only responsible for the task assigned by the master node and receives the assignment from the master node, the slave node starts to find suitable computing nodes for its subordinate storage nodes [11]. First, the slave node is responsible for the task assigned by the master node. The computing resource margin of the node is detected. If the remaining computing resources are sufficient to satisfy the user's job submission, the computing resources of the node will be allocated first. If the resources have been exhausted or insufficient to meet the minimum computing resources promised to the user, the search for other suitable computing resources in the cloud environment will begin. The improved Ant colony allocation algorithm introduced in this paper will search within a certain range in order to reduce the network overhead [12]. If there is still no suitable resource, the user data mirror fragments in the cluster will be removed from the main job scheduling node.

### 3.2 Evaluation Condition of Calculating Resource's Merits and Demerits

Consider the node domain as an undirected graph *G*(*V*, *E*), where *V* is the set of all nodes in region Area, and *E* is the set of networks connecting the nodes. Find the appropriate compute node, that is, find an optimal path in *E*. The metrics can consider the following factors:

(1) Estimated execution time: time_cost(*e*), which refers to the estimated time taken by the computing resource at the end of path e to process the operation.

(2) Network bandwidth: bandwith(*e*) refers to the maximum bandwidth of the network provided by path e.

(3) Network delay: delay(*e*), which refers to the maximum network delay generated by path *e* Resource selection constraint function:

$$res(e) = \frac{Atime\_cost(e) + Cdelay(e)}{Bbandwith(e)} \tag{1}$$

$$\text{s.t.} \begin{Bmatrix} time\_cost(e) < TL \\ bandwith(e) > EL \\ delay(e) < DL \end{Bmatrix} \tag{2}$$

The process of selecting resources and paths is to find the path and resources of the res(*e*) that satisfy the constraint (2) as much as possible. *A*, *B*, and *C* are the weights of the three constraints; *TL*, *EL*, and *DL* are the boundary constraints. Different cloud computing environments may have different values.

### 3.3 Predict the Execution Speed of Each Task by Computing Resources

Heterogeneity is a very important feature in the whole cloud environment. That is to say, the structure, hardware and software environment, capacity and throughput of each node will be different. At the same time, the network situation is more complex, and the load of any line at any time will be unpredictable, because the network bandwidth in cloud computing environment is much larger than that in other environments. The traditional grid environment has low bandwidth, and the network situation will change dramatically and unpredictably. Therefore, it will be very difficult to calculate the execution speed of the nodes. However, in cloud computing, assigning tasks to the most efficient and least expensive computing resources will greatly improve the overall performance. In view of the heterogeneous and changing characteristics of cloud computing, through the prediction algorithm in document [8], a prediction model is designed to estimate the next task execution speed by accumulating historical values. The model predicts the efficiency and time of each node to complete the next task separately. It is hoped that no matter how much computing resources are loaded a relatively accurate prediction can be obtained by this model. Since the current load level of each computing node is known and the average load level at the last job completion can be consulted, we use the following models to predict the execution speed:

$$EV_m^{a_{k+1}}(k+1) = \frac{a_{k+1}}{a_k}\left((1-\rho)EV_m^{a_k}(k) + \rho RV_m^{a_k}(k)\right) \tag{3}$$

where $EV_m^{a_k}(k)$ is the kth predicted execution speed of the *m*-th computing resource, the unit can be MIPS, which is the system load degree at the kth prediction, $RV_m^{a_k}(k)$ refers to the *m*-th actual kth execution speed, and $\rho$ is an adjustment. The parameter is used to adjust the proportion of the empirical value and the predicted value in different cloud environments to optimize the prediction of the model. On each computing node, the node itself will record the completion of the job each time a job is completed. The actual speed is combined with the last prediction results to estimate the possible execution speed of the next job. At the same time, the system load is recorded together. This is an important parameter; generally, there are several quantitative indicators, such as the actual usage of the CPU, the number of jobs or the number of threads [13].

### 3.4 Use Improved Ant Colony Algorithm to Find the Most Suitable Computing Resources

Because in the cloud computing environment the specific situation of the resource is unknown and the network does not have a fixed topology, the structure and resource distribution of the entire cloud environment is unpredictable and its actual situation is unforeseeable too. In this case, the location of the computing resource and quality is unknown to data nodes [14].

Using the improved Ant colony algorithm, it is possible to find computing resources in an unknown network topology and select the most appropriate one or several assignments to the user until the user requirements are met. When the search starts, the slave node issues a query message. It plays the role of ant in the improved Ant colony algorithm [15]. All ants follow the pheromone node with high probability, and the pheromone with less node probability chooses the next hop node and leaves the pheromone on the passing path node.

(1) Setting for the next hop selector

In the initial state, each link pheromone content in the grid structure is the same.

Setting: $d$ time, the probability of each node in next hop is calculated under the ant following formula of the node:

$$j=\begin{cases}\text{if, } q<q_0,\ \max_{n\notin avids(k)}\left\{\dfrac{\tau_{in}(d)^{a}\left(EV_n^{a_d}(d)\right)^{\gamma}}{L_{q_{in}}(d)^{\beta}}\right\}\\ \text{else, Determine the probability of adjacent nodes selected by formula}\end{cases}\tag{4}$$

$$p_{ij}^{k}=\begin{cases}\text{if}, j\notin avid(k), \dfrac{\left\{\dfrac{\tau_{in}(d)^{a}(EV_j^{a_d}(d))^{\gamma}}{L_{q_{ij}}(d)^{\beta}}\right\}}{\left\{\sum\limits_{n\notin avid}\dfrac{\tau_{in}(d)^{a}(EV_n^{a_d}(d))^{\gamma}}{(L_{q_{in}}(d))^{\beta}}\right\}}\\ \text{else, } 0.\end{cases}\tag{5}$$

where $\tau_{in}(d)$ is the time $d$, the forward ant $i$ observes the pheromone intensity of $j$ nodes on the node, $p_{ij}^{k}$ is the probability that the ant chooses $j$ points at $i$ o'clock, $avid(k)$ is the avoidance list of ant $k$, and $L_{q_{ij}}=\dfrac{Cdekat(e_{ij})}{Bbandwith(e_{ij})}$ is from $i$ nodes to $j$-node line quality, $\alpha$, $\beta$ and $\gamma$ is the relative weight of the pheromone, line quality and computing power predictions.

In order to prevent the result from converging too quickly on the local optimal solution, the random coefficients $q\in[0,1]$ and $q_0\in[0,1]$ are set. These two values and formulas are used to control the probability that the ant directly selects the pheromone-line quality ratio to the largest adjacent node.

(2) Pheromone update

In order to reflect the change of pheromone in time, use the local update strategy to modify the pheromone strength on the node. All nodes with non-zero pheromone are updated with time as follows:

$$\tau_{ij}(t+1)=\rho\times\tau_{ij}(t)\tag{6}$$

If the node generates or receives a backward ant within the waiting time period of this update, it is updated according to the following formula:

$$\tau_{ij}(t+1)=\rho\times\tau_{ij}(t)+(1-\rho)D\frac{1}{\sum res(e_{ij})}\tag{7}$$

wherein, constant $\rho$ is the residual coefficient of the pheromone, $\tau_{ij}(t+1)$ means that the pheromone intensity on the $j$ nodes is seen from the node $i$ at time $t$ + 1, and when the value is less than a small value of $m$, $\tau_{ij}$ is zeroed so that the ant can converge faster. Go to the path with the appropriate resources, and expand the pheromone strength of the nodes on the path with the appropriate resources according to its resource constraint function. $D$ is the effect expansion coefficient [16, 17].

## 4 ALGORITHM SIMULATION

MATLAB software is used to simulate the running environment of cloud computing to randomly generate resource parameters of network nodes and network nodes to achieve the purpose of real simulation of network environment. Check the operation of the algorithm in this special network environment, in order to build a comparatively realistic cloud environment locally through the settings [18]. In the external environment it is the same, through the adjustment of parameters to detect the performance of this algorithm.

Among them, the setting of the algorithm is shown in Tab. 1.

**Table 1** Algorithm parameter settings

| Parameter | Value |
|---|---|
| Number of nodes | 100 |
| Maximum algebra | 20 |
| Ant number | 60 |
| $\alpha$ | 1 |
| $\beta$ | 5 |
| $\rho$ | 0.1 |
| $Q$ | 100 |
| Cpumax | 8 |
| Fixcpu | 5 |

Simulation results: Fig. 1 shows that there are 100 randomly generated network nodes in the network. The parameters of the network nodes are assumed to be based on the CPU of each node, and can also be set to memory, CPU usage, network bandwidth, and so on. Fig. 1 shows that 40% of the nodes are searched for when the defined number of programs is terminated, and roughly converge in the 8th generation.

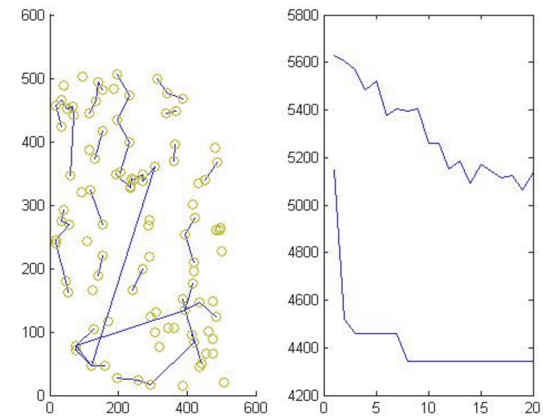


**Figure 1** finds that 40% of the nodes reach the maximum algebra stop

Fig. 2 shows that 40% of the nodes are searched for a stop. As can be seen from the figure, the search for 3 generations has reached the requirement. Comparing 1 and 2, the right resource should be found in the network environment as the fastest priority, so the situation in Fig. 2 should be used, so that the resources that meet the needs can be quickly found and distributed to the corresponding nodes for processing.

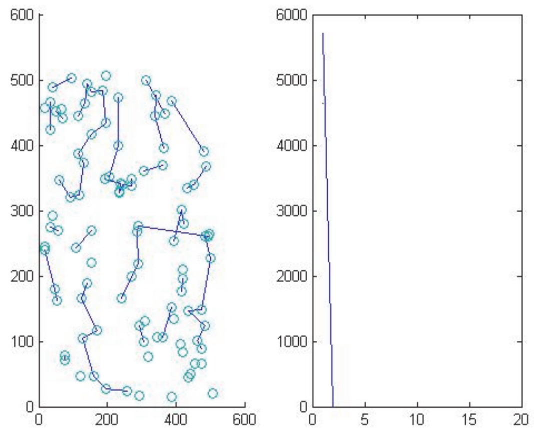


**Figure** 2 Searched 40% of the nodes searched to stop

## 5 CONCLUSION

The energy structure in the world is changing. The development of new energy industry is an inevitable trend of the world energy development, but also a necessary condition for China to comprehensively deepen the reform in the energy field, and then realize the economic transformation [19, 20]. In terms of operating costs, cloud computing has lower requirements for the network environment than the grid environment, and the service content such as SAAS, PAAS and IAAS provides higher requirements than the grid to the infrastructure, so it is a good resource scheduling The algorithm is especially important. This paper has made a series of improvements to the ant colony algorithm. Then MATLAB simulation experiments are carried out. The experimental results show that the task completion time is shorter, and the improvement is more effective when the number of tasks is large, indicating that the improved algorithm is more suitable for large-scale resource allocation problems. The improved Ant colony resource allocation algorithm introduced in this paper can dynamically search and allocate computing resources for users' operations according to the large-scale, sharing and dynamic characteristics of the cloud environment. From the simulation results, it can be seen that the optimization algorithm can effectively complete the search and allocation of computing resources in the cloud computing environment. It has important practical and theoretical significance which can optimize the allocation of factors, improve productivity and promote industrial development.


### Acknowledgements

This work was supported by the National Social Science Foundation of China (Grant number 18ZDA086).

The authors would like to express their sincere thanks to the anonymous reviewers for their constructive comments on an earlier version of this manuscript.


## 6 REFERENCES


[1] Prem Jacob, T. & Pradeep, K. (2019). A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. *Wireless Personal Communications*, *109*(1), 315-331. https://doi.org/10.1007/s11277-019-06566-w

[2] Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *Ieee Access*, *3*, 2687-2699. https://doi.org/10.1109/ACCESS.2015.2508940

[3] Mohajerani, A. & Gharavian, D. (2015). An ant colony optimization based routing algorithm for extending network lifetime in wireless sensor networks. *Wireless Networks*, *22*(8), 1-11. https://doi.org/10.1007/s11276-015-1061-6

[4] Mansouri, N., Zade, B. M. H., & Javidi, M. M. (2020). A multi-objective optimized replication using fuzzy based self-defense algorithm for cloud computing. *Journal of Network and Computer Applications*, *171*, 102811. https://doi.org/10.1016/j.jnca.2020.102811

[5] Xie, X., Xu, K., & Wang, X. (2019). Cloud computing resource scheduling based on improved differential evolution ant colony algorithm. *In Proceedings of the 2019 International Conference on Data Mining and Machine Learning (pp. 171-177)*. https://doi.org/10.1145/3335656.3335706

[6] Nie, Q., & Li, P. (2016, October). An improved ant colony optimization algorithm for improving cloud resource utilization. *In 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 311-314). IEEE.* https://doi.org/10.1109/CyberC.2016.66

[7] Kumar, A., & Bawa, S. (2019). Generalized ant colony optimizer: swarm-based meta-heuristic algorithm for cloud services execution. *Computing, 101*(11), 1609-1632. https://doi.org/10.1007/s00607-018-0674-x

[8] Meshkati, J., & Safi-Esfahani, F. (2019). Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing. *The Journal of Supercomputing, 75*(5), 2455-2496. https://doi.org/10.1007/s11227-018-2626-9

[9] Tong, Z., Deng, X., Ye, F., Basodi, S., Xiao, X., & Pan, Y. (2020). Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment. *Information Sciences, 537*, 116-131. https://doi.org/10.1016/j.ins.2020.05.057

[10] Gonçalves, G. E., Endo, P. T., Rodrigues, M., Sadok, D. H., Kelner, J., & Curescu, C. (2020). Resource allocation based on redundancy models for high availability cloud. *Computing*, *102*(1), 43-63. https://doi.org/10.1007/s00607-019-00728-1

[11] Zhu, L., Li, Q., & He, L. (2012). Study on cloud computing resource scheduling strategy based on the ant colony optimization algorithm. *International Journal of Computer Science Issues (IJCSI)*, *9*(5), 54.

[12] Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2022). A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques. *Sensors*, *22*(3), 1242. https://doi.org/10.3390/s22031242

[13] Shang, Q. (2021). A dynamic resource allocation algorithm in cloud computing based on workflow and resource clustering. *Journal of Internet Technology*, *22*(2), 403-411. https://10.3966/160792642021032202015

[14] Dakshayini, D. M. & Guruprasad, D. H. (2011). An optimal model for priority based service scheduling policy for cloud computing environment. *International journal of computer applications*, *32*(9), 23-29.

[15] Chen, J., Wang, Y., & Liu, T. (2021). A proactive resource allocation method based on adaptive prediction of resource

requests in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), 1-20. https://doi.org/10.1186/s13638-021-01912-8
[16] Chen, J., Du, T., & Xiao, G. (2021). A multi-objective optimization for resource allocation of emergent demands in cloud computing. *Journal of Cloud Computing*, *10*(1), 1-17. https://doi.org/10.1186/s13677-021-00237-7
[17] Belgacem, A. (2022). Dynamic resource allocation in cloud computing: analysis and taxonomies. *Computing*, *104*(3), 681-710. https://doi.org/10.1007/s00607-021-01045-2
[18] Su, Y., Bai, Z., & Xie, D. (2021). The optimizing resource allocation and task scheduling based on cloud computing and Ant Colony Optimization Algorithm. J*ournal of Ambient Intelligence and Humanized Computing*, 1-9. https://doi.org/10.1007/s12652-021-03445-w
[19] Gao, L., Zhang, X., Tong, T., & Zhang, J. (2018, September). Research on new energy utilization based on cloud computing and big data. *2018 7th International Conference on Energy and Environmental Protection (ICEEP 2018)*, 371-375. https://doi.org/10.2991/iceep-18.2018.64
[20] Jing, L., Liu, G., Guo, X., & Su, S. (2022). Research on the cloud computing fuzzy proportion integration differentiation control strategy for permanent-magnet homopolar motor with salient pole solid rotor used on new-energy vehicle. *Sustainable Energy Technologies and Assessments*, *52*, 101969. https://doi.org/10.1016/j.seta.2022.101969

**Contact information:**

**Haoyang DU**, PhD
School of Economics and Management Beijing Jiaotong University,
Beijing 100044, China
Henan University of Economics and Law,
Zhengzhou 450046, China
E-mail: haoyangdu2008@163.com

**Junhui CHEN**, PhD
(Corresponding author)
School of Economics and Management Beijing Jiaotong University,
Beijing 100044, China
Henan University of Economics and Law,
Zhengzhou 450046, China
E-mail: junhuichen163@163.com