Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2022.030301
ISSN 2790-0916 Vol. 3, Issue 3: 1-8

Distributed System Task Assignment Considering Ant Colony Algorithm

Jian Yang*

College of Electronic Engineering, National University of Defense Technology, Hefei 230031, Anhui, China
yangjian_eei@163.com
**corresponding author*

Keywords: Ant Colony Algorithm, Distributed System, Task Allocation, Resource Scheduling

Abstract: In a company of a certain scale, a large number of tasks need to be allocated every day. It is particularly important to provide a comprehensive and efficient distributed system for resource scheduling. Ant colony algorithm (ACO) has great advantages in resource scheduling, so In this paper, the problem of task assignment in distributed systems is studied considering the ant colony optimization algorithm. The structure of this paper can be divided into three parts, including related theoretical overview, system design and system analysis. In the system design and system analysis, through the comparative analysis of the ant colony algorithm (ACO), the ant colony optimization algorithm based on load balancing (LBACO) and the improved ant colony algorithm (IACO), the optimal algorithm is found. Assign tasks.

1. Introduction

With the rapid development of computer and information technology and its in-depth integration with traditional technologies in various industries, the data transmitted by task allocation is huge, which will inevitably increase the load pressure of distributed systems. An efficient and reasonable resource scheduling scheme will meet the needs of users At the same time of demand, resources can also be used to the fullest, avoiding the situation that resources are wasted due to idleness or time-consuming due to waiting [1-2]. ACO was initially used to solve the traveling salesman problem by virtue of its good performance in finding optimal paths, and was gradually used in solving task scheduling problems [3].

At present, many researchers have carried out in-depth research on ACO and task allocation in distributed systems, and have achieved good results. For example, scholars such as Dnmez E proposed an intelligent planning (IP) system for scenic routes based on ACO. The software part of

the system uses ACO to convert the landscape route planning problem into the shortest feasible route problem, and calculates the transition probability of ants to scenic spots (SS) by calculating the transition probability of ants to SS. Construct the weight matrix path search used by ants in this process, build an IP model, and realize the IP of SS. The results show that the system can solve the problems existing in the traditional routing planning system [4]. Researchers such as Abhijit A proposed an adaptive variable neighborhood search ACO to solve the vehicle routing problem with soft time window. The validity of the problem, and the comparative analysis of the experimental results of the two algorithms shows the advantages of the IACO. The experimental results show that the proposed algorithm can effectively obtain better solutions [5]. However, there are relatively few researches on the task assignment of distributed systems based on ACO. ACO is very advantageous in finding optimal paths. Therefore, it is a worthy direction to study distributed systems by using ACO. Assignment of tasks is very beneficial.

In this paper, ACO is used to study the problem of task allocation in distributed systems. The structure of this paper can be divided into three parts: the first is to introduce the task allocation model and the goal of task allocation; The number of iterations and the number of iterations are researched and designed. Finally, the task execution completion time and task execution cost are analyzed. Through the analysis, it is found that the IACO algorithm has more advantages in task completion time and cost.

2. Related Overview

2.1. Task Assignment Model

In a broad sense, task allocation refers to selecting a suitable running virtual machine for the task uploaded by the user to achieve the maximum utilization efficiency of the system under the constraint of user service quality [6]. Designing a suitable task scheduling algorithm in a suitable virtual machine environment helps to allocate computing resources reasonably, thereby reducing computing costs, improving task processing efficiency, and speeding up user feedback responses [7-8]. The process of task assignment is shown in Figure 1.

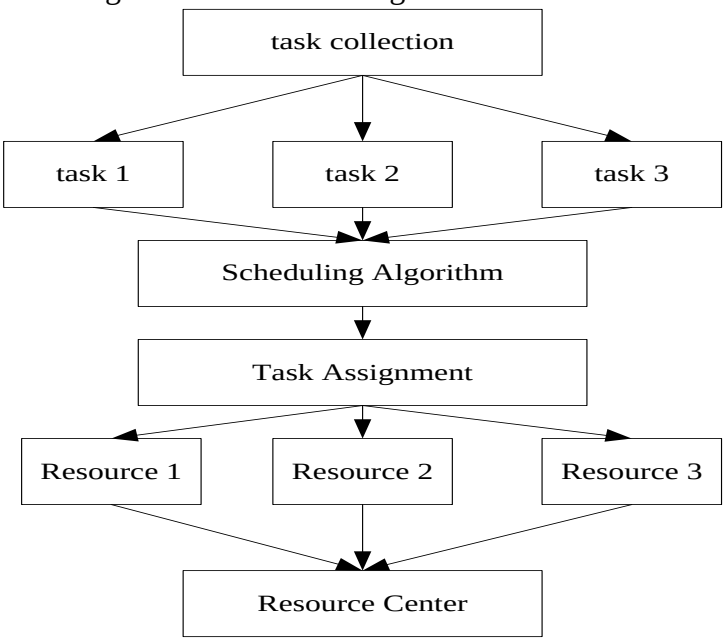

Figure 1. Task scheduling flowchart

In the process of task allocation, the scheduling algorithm generally divides the tasks, which is very complicated in practice [9]. In order to simplify the process of researching the algorithm, this paper idealizes some realistic constraints: tasks are independent of each other, the execution of tasks is not sequential and cannot be interrupted or migrated, and the computing power of all resources is known.

2.2. Task Assignment Goals

Task allocation has the following three goals, namely optimal span, load balancing and economic principles, and the following is a detailed description of the above three goals.

(1) Optimal span. That is to minimize the time taken by the user-submitted tasks from importing the platform to the completion of all tasks and returning the corresponding results to the user [10].

(2) Load balancing. Considering that the computing power of each node and the computing load required by each task are difficult to achieve a high degree of adaptation at low cost during the implementation process of the system, the process of achieving a balanced distribution of the load will greatly affect the computing efficiency [11].

(3) Economic principles. The further development of distributed system task allocation problem not only depends on the vital interests of users, but also depends on the economic benefits of service providers to a certain extent. Maximizing the interests of both parties in a balanced state is also an important task goal of resource scheduling [12-13].

3. System Design

3.1. Pheromone Correction Coefficient

Regarding the pheromone correction coefficient α proposed in this paper, after each ant completes a search process, the pheromone will be based on the length of the optimal path obtained by the ant this time [14]. The multiple difference (MD) between the optimal solution and the worst solution is:

$$(1+\alpha)-(1-\alpha)=2\alpha \tag{1}$$

The MD between the optimal solution and the general solution is:

$$(1+\alpha)-(1-\alpha^2)=\alpha+\alpha^2 \tag{2}$$

We hope that the MD 2α between the optimal solution and the worst solution is as large as possible, try to widen the distance between the two, and maximize the attractiveness of the optimal solution, so that the ants in the future will choose more Short path, and minimize the interference of the worst solution, so that it is no longer selected by subsequent ants; at the same time, we also hope that the MD α +α2 between the optimal solution and the general solution should not be too large as much as possible, keep it The attractiveness of the intermediate solution gives the ants more choices, preventing the following ants from choosing the optimal solution, narrowing the scope of optimization, and falling into the local optimal solution.

Formula (3) is obtained by subtracting formula (1) and formula (2):

$$2\alpha-(\alpha+\alpha^2)=\alpha-\alpha^2 \tag{3}$$

Formula (4) is the result of derivation of formula (3) so that the result obtained is as large as possible, namely:

$$(\alpha - \alpha^2)' = 1 - \alpha \tag{4}$$

Let the result of the formula be 0, we can get the maximum value of formula (4) when α=0.5. Therefore, the value of the pheromone correction coefficient α is temporarily 0.5 under ideal conditions.

3.2. Load Balancing

In order to evaluate the load balancing effect of different algorithms, this paper proposes load disparity (DI) to evaluate the load performance of the distributed system after the task scheduling is completed. DI is defined as:

$$DI = \frac{E_{\max} - E_{\min}}{E_{avg}} \tag{5}$$

It can be seen from equation (5) that the smaller the DI value, the lower the load unevenness, indicating that the load balancing degree of the system is better; otherwise, the load balancing performance of the system is worse [15-16]. The load unevenness DI value of the three algorithms ACO, LBACO, and IACO is shown in Figure 2 when processing 100 to 500 tasks.

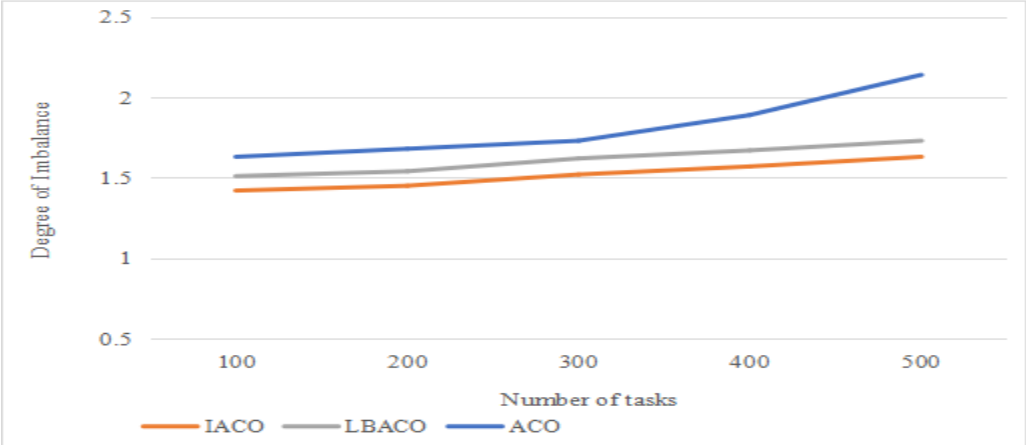

Figure 2. Load imbalance diagram

As can be seen from Figure 2, IACO and LBACO both maintain relatively stable and low DI values when performing different numbers of tasks, while the DI value of ACO is relatively high and fluctuates greatly with the change of the number of tasks. In the three algorithms IACO has the lowest DI value. This shows that IACO has certain advantages in system load balancing compared with LBACO and ACO, and the introduction of the virtual machine evaluation factor has played a better role in improving system load balancing. It can be seen from Figure 2 that both IACO and LBIAO proposed in this paper have excellent performance in system load balancing, and the average performance of IACO and LBTIAO is better than that of ACO. The introduction of the pheromone correction coefficient a enables IACO to perform tasks in less time in most cases, and the introduction of the virtual machine evaluation factor enables IACO to ensure the load balance of the system. These results prove the effectiveness and feasibility of IACO sex.

3.3. Design of the Number of Iterations

In experiments evaluating the effect of iterations on convergence, the default pheromone factor was equal to 1, the heuristic expectation factor was equal to 1, and the pheromone volatility factor

was equal to 0.5; the number of ants was 10. Figure 3 shows the effect of the number of iterations on the convergence results.

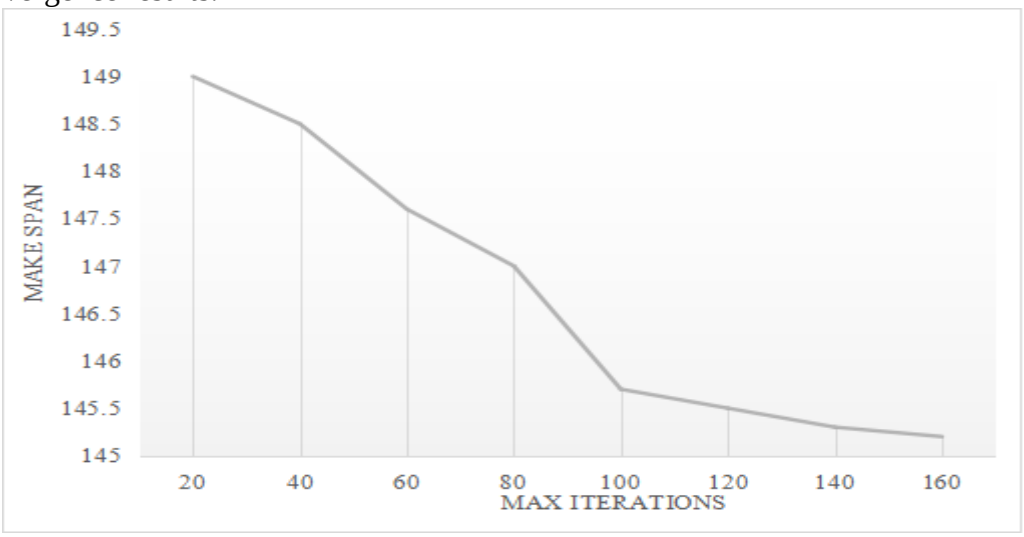

Figure 3. Iteration number graph

The maximum number of iterations affects the convergence results of the algorithm. The more iterations, the greater the possibility of the ant colony finding the global optimal solution [17]. It can be seen from Figure 3 that with the iterative cycle of the algorithm, the final shortest task execution time of the ant colony on the virtual machine gradually decreases. When the maximum number of iterations is equal to 100, the downward trend tends to be gentle, and the convergence result of the algorithm does not occur anymore. The obvious change indicates that the ant colony is approaching to obtain the global optimal solution. Therefore, this paper sets the maximum number of iterations to 100.

4. System Analysis

Set the number of virtual machine resources in the distributed system to remain unchanged. Here, the number of tasks is taken as 100, 200, 300, 400, and 500 respectively. The test compares the impact of the number of tasks on the task completion time and cost under the three algorithms of ACO, IACO and LBACO. The parameter settings of this round of experiments are shown in Table 1. The experiment obtains the changes of the total task completion time and the total cost of the task with the number of tasks. Figure, as shown in Figure 4 and Figure 5.

Table 1. Parameter configuration table

Parameter name	Parameter value
Number of tasks	100,200,300,400,500
Number of resources	100,50
Iterations	100

4.1. Analysis of Task Completion Time

The purpose of this experiment is to verify whether the three algorithms reduce the time span of executing tasks during resource allocation. The experimental parameters are shown in Table 1, and

the time spans of executing tasks at different levels are calculated respectively. The experimental results are shown in Figure 4.

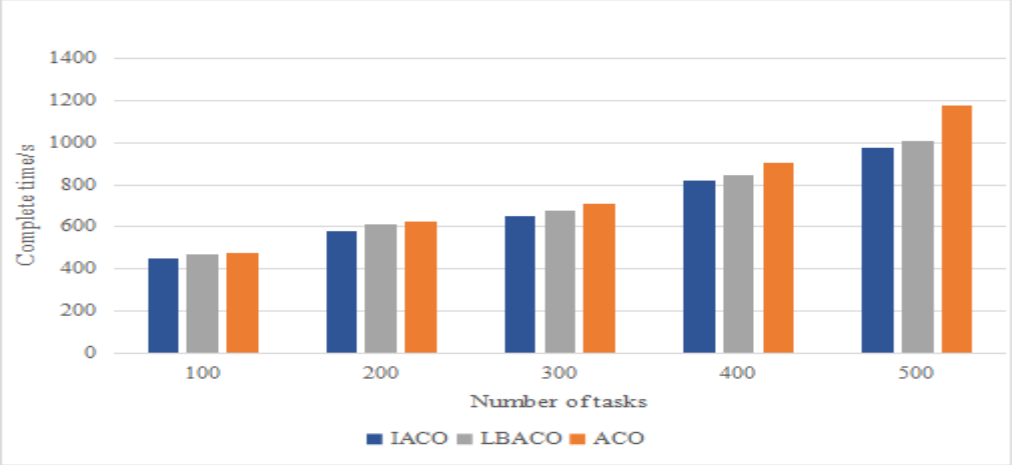

Figure 4. Completion time comparison chart

As can be seen from Figure 4, as the number of tasks increases, the completion time of the three scheduling algorithms increases, but the ACO task completion time is higher than the other two algorithms; when the number of tasks is the same, the ACO has the largest completion time The completion time of the IACO and the LBACO is slightly different, but overall the completion time of the IACO is lower than that of the LBACO. In terms of completion time, the IACO and the LBACO are significantly better than the ACO algorithm. The difference between the completion time of IACO and LBACO is relatively small. The main reason is that the two algorithms have a time constraint function that acts on the completion time. Although the form of the constraint function is different, they are both used to redefine the pheromone of the ACO. There is consistency in the way of action, so that the completion time of the two algorithms is smaller than that of the ACO, and the difference between the IACO and the LBACO in the completion time is small, but on the whole, the completion time of the IACO is lower than that of the LBACO.

4.2. Task Cost Analysis

In actual task allocation, each virtual machine has the property of cost per unit time, so the total cost can be calculated according to the execution time of each user task on the virtual machine and the cost per unit time [18]. The purpose of this experiment is to verify whether the cost of resource scheduling of the three algorithms is reduced. During the experiment, under the condition that the configuration of each entity is kept consistent, the execution cost of different tasks is counted separately. The experimental results are shown in Figure 5.

From the results in Figure 5, the following conclusions can be drawn: the cost of the three algorithms increases as the number of tasks increases, but the ACO algorithm consumes the most cost; when the number of tasks is the same, the IACO algorithm Compared with the ACO algorithm, the cost advantage of the IACO algorithm in resource scheduling is mainly due to the addition of a cost constraint function to the algorithm, which incorporates the price of the virtual machine into the scheduling factor. Among them, when the time cost function is used to redefine the pheromone, the higher the cost of the virtual machine, the less pheromone left on the virtual machine, and the smaller the probability of selecting the virtual machine. The higher the probability of the machine, the ant colony tends to select the virtual machine node with high probability, that is, the virtual

machine with low cost is easier to be selected. Therefore, the algorithm effectively reduces the cost of executing tasks.

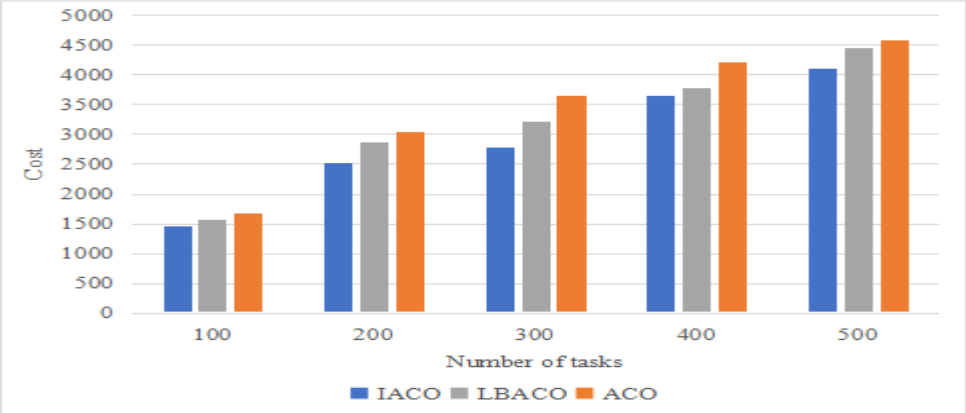

Figure 5. Cost comparison chart

5. Conclusion

Based on the advantages of the ACO in optimizing the path, this paper uses the ant colony optimization algorithm to study the task assignment problem of the distributed system. In this paper, the three algorithms of ACO, IACO and LBACO are compared and analyzed in terms of load balancing, task completion time and task cost. Through comparative analysis, it is found that the IACO algorithm has more advantages in system load balancing, with the shortest task completion time and the lowest task cost, followed by the LBACO algorithm, and the ACO algorithm has the worst system load balance, the longest task completion time and the highest cost. This paper studies the task assignment of distributed systems on the basis of ACO, but due to its limited ability, there are still many shortcomings in the study of task assignment, which need to be improved. Taking into account the ACO to study the distributed system is a direction worthy of in-depth discussion.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Khandelwal A. Fuzzy based Amalgamated Technique for Optimal Service Time in Distributed Computing System. International Journal of Recent Technology and Engineering, 2019, 8(3):6763-6768. https://doi.org/10.35940 /ijrte.C4783.098319

[2] Satheeshkumar A D. Public auditing and energy saving task scheduling strategy based on round robin algorithm in cloud computing. Journal of Advanced Research in Dynamical and Control Systems, 2018(12):44-49.

[3] Rahimi-Farahani H, Rassafi A A, Mirbaha B. Forced-node route guidance system: incorporating both user equilibrium and system optimal benefits. IET Intelligent Transport Systems, 2019, 13(12):1851-1859. https://doi.org/10.1049/iet-its.2018.5457

[4] Dnmez E, Kocamaz A F. oklu Hedeflerin oklu Robotlara Paylatrlmas in Bir Yük Dengeleme Sistemi. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 2019, 8(2):533-548. https://doi.org/10.17798/bitlisfen.467757

[5] Abhijit A, Sulabha S. Performance Enhancement of Distributed System through Load Balancing and Task Scheduling. International Journal of Computer Applications, 2018, 181(3):20-26. https://doi.org/10.5120/ ijca2018917391

[6] Jzab C, Jxa B. Cooperative task assignment of multi-UAV system. Chinese Journal of Aeronautics, 2020, 33(11):2825-2827. https://doi.org/10.1016/j.cja.2020.02.009

[7] Anselmi J, Doncel J. Asymptotically Optimal Size-Interval Task Assignments. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(11):2422-2433. https://doi.org/10.1109/TPDS.2019.2920121

[8] Kanso B, Kansou A, Yassine A. Open Capacitated ARC routing problem by Hybridized Ant Colony Algorithm. RAIRO - Operations Research, 2021, 55(2):639-652. https://doi.org/10.1051/ro/2021034

[9] Euchi J, Sadok A. Optimising the travel of home health carers using a hybrid ant colony algorithm. Transport, 2021(3):1-22. https://doi.org/10.1680/jtran.19.00114

[10] Al-Amyal F, Hamouda M, L Számel. Torque Quality Improvement of Switched Reluctance Motor Using Ant Colony Algorithm. Acta Polytechnica Hungarica, 2021, 18(7):129-150. https://doi.org/10.12700/APH.18.7.2021.7.7

[11] Kanso B, Kansou A, Yassine A. Open Capacitated ARC routing problem by Hybridized Ant Colony Algorithm. RAIRO - Operations Research, 2021, 55(2):639-652. https://doi.org/10.1051/ro/2021034

[12] Lima V, Lima E, Sherafat H. Roteirization of vehicles in the delivery/collection problems - Application of a modifed Ant Colony Algorithm. Revista Brasileira de Computação Aplicada, 2020, 12(1):44-53. https://doi.org/10.5335/rbca.v12i1.9317

[13] Srinivasan R, Jayaraman M. Experimentation on product and service life cycle On drive shaft using ant colony algorithm. Journal of the Balkan Tribological Association, 2020, 26(4):729-735.

[14] Olkhova M, Roslavtsev D, Matviichuk O, et al. City Delivery Routes Planning Based on the Ant Colony Algorithm. Science & Technique, 2020, 19(4):356-362. https://doi.org/10.21122/2227-1031-2020-19-4-356-362

[15] Sakir R, Bhardwaj S, Kim D S. Enhanced faulty node detection with interval weighting factor for distributed systems. Journal of Communications and Networks, 2021, 23(1):34-42. https://doi.org/10.23919/JCN.2021.000002

[16] Saraswat B K, Suryavanshi R, Yadav D. Formal Specification & Verification of Checkpoint Algorithm for Distributed Systems using Event - B. International Journal of Engineering Trends and Technology, 2021, 69(4):1-9. https://doi.org/10.14445/22315381/IJETT-V69I4P201

[17] Santos A A, Silva A, Magalhes A P, et al. Determinism of Replicated Distributed Systems-A Timing Analysis of the Data Passing Process. Advances in Science Technology and Engineering Systems Journal, 2020, 5(6):531-537. https://doi.org/10.25046/aj050663

[18] Abdukarimovich G N, Khudainazarovna K M, Ravshabekovna S S. Building models of territorial distributed systems. International Journal on Integrated Education, 2020, 3(10):300-303. https://doi.org/10.31149/ijie.v3i10.762