Research Article

Cloud Service Optimization Method Based on Dynamic Artificial Ant-Bee Colony Algorithm in Agricultural Equipment Manufacturing

Kai Zhou,1 Yongzhao Wen,2 Wanying Wu,1 Zhiyong Ni,1 Tianguo Jin,3 and Xiaojun Long1

1College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai’an 271018, China
2Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110169, China
3College of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Xiaojun Long; longxjsdau@163.com

Received 6 June 2020; Accepted 30 September 2020; Published 28 October 2020

Abstract

In view of the miniaturization and decentralization characteristics of agricultural equipment factories in China, agricultural equipment manufacturing is well suited to the cloud manufacturing model, but there is no specific research on cloud services optimization for it. To fill the research gap, a cloud service optimization method is proposed in this paper. For the optimization model, the dynamic coefficient strategy and the reliability feedback update strategy are added to the mathematical model to strengthen the applicability of farming season. As optimization algorithm, a dynamic artificial ant-bee colony algorithm (DAABA) based on artificial ant colony algorithm and bee colony algorithm is presented. The optimal fusion evaluation strategy is used to save optimization time by reducing the useless iteration, and the iterative adjustment threshold strategy is adopted to improve the accuracy of cloud service by increasing the size of bee colony. Finally, the performance of DAABA is verified to be more superior by comparing with other algorithms.

1. Introduction

At present, the distribution of agricultural equipment factories is characterized by decentralization and miniaturization in China, which causes unbalanced equipment utilization in agricultural equipment manufacturing. Many small-scale agricultural equipment factories are still in the workshop production stage with low production efficiency and chaotic management [1]. In order to improve production efficiency, those small-scale agricultural equipment factories have started to organize the production through cloud service platforms [2]. However, massive nonstandard components that need to be produced on the spot are used in the manufacturing process of agricultural equipment and their quality is difficult to be quantified, which causes the accuracy and efficiency of cloud service in agricultural equipment to be hard to improve.

Cloud manufacturing is an integrated product of advanced information technology, manufacturing technology and emerging Internet of things technology [3]. In cloud manufacturing, the processing information of the factory is virtualized and digitized, which is encapsulated as a cloud service pool to share resources for individual users [4, 5]. In the process of cloud manufacturing, the manufacturing task is decomposed into several subtasks, and the corresponding cloud service assemblage is matched for each subtask; then the concrete service is selected. Therefore, many factories cooperate to accomplish the manufacturing task, and the processed product needs to be continuously transported to new factory for further processing in accordance with the manufacturing procedure. The longer the distance is, the higher the cost is, so the transport cost should be taken into consideration as a TSP problem [6]. There is obvious mutual restriction among the subtasks, so optimization of subtasks has become a multidimensional problem, which limits the precision and efficiency of cloud service.

At present, artificial intelligence algorithms have been widely used in complicated optimization problems because
of their fast optimization, such as genetic algorithm (GA) [7], particle swarm optimization (PSO) [8], MAX-MIN ant system (MMAS) [9,10], artificial bee colony (ABC) algorithm [11], and firefly algorithm [12]. Kumar and Bawa presented a generalized ant colony optimizer (GACO) algorithm [13], but GACO was not tested in the concrete model, so the practicability of GACO needed to be explored. Ghomi et al. applied a genetic algorithm to achieve scheduling and logistics optimization in cloud manufacturing [14], but there were some limitations in the choice of algorithm. Duan et al. presented an adaptive incremental genetic algorithm for task optimization in cloud environments [15], but there was plenty of room for improvement. Yang et al. used dynamic ant colony genetic algorithm to achieve optimization in cloud service [16], but the TSP problem of transport cost was not considered in detail. Due to the slow developments of agricultural equipment, there is no specific optimization method in cloud service. Thus, the optimization model and algorithm for cloud services need to be further studied in agricultural equipment manufacturing to improve the production efficiency.

With developments of agriculture in China, more and more pieces of equipment are used in agricultural production, so it is necessary to improve the accuracy and efficiency of cloud service optimization in agricultural equipment manufacturing. Therefore, a new cloud service optimization method based on dynamic artificial ant-bee colony algorithm is proposed in this paper. Firstly, the dynamic coefficient strategy and reliability feedback update strategy are applied to the Quality of Service (QoS) evaluation model to enhance the applicability of mathematical model in busy farming. Then, DAABA is designed based on artificial ant colony algorithm and bee colony algorithm. Furthermore, traditional artificial bee colony algorithm is added to mutation operation to improve the accuracy of optimal solution, and the optimal fusion evaluation strategy are adopted to improve the accuracy and efficiency of cloud service in agricultural equipment manufacturing. Finally, the convergence of DAABA is verified through theoretical derivation and simulation; then the accuracy and efficiency of DAABA are verified to be higher than GA, ABC, and MMAS through experimental simulation.

2. Mathematical Model

2.1. Cloud Service Optimization Process in Agricultural Equipment Manufacturing. Due to the strong seasonal characteristic of agricultural production, the manufacturing tasks of agricultural equipment tend to be concentrated in busy farming. In the context of large-scale orders, a reasonable scheme can significantly improve the efficiency of factory [17]. So, it is necessary to analyze the process of cloud manufacturing. As shown in Figure 1, the manufacturing task is decomposed into many subtasks, and the corresponding cloud service assemblage is matched for each subtask; then the optimal factory is selected from corresponding cloud service in a certain order, so the optimal manufacturing scheme is determined. The specific process can be divided into the following steps:

1. Manufacturing task decomposition: through the cloud platform, the manufacturing task of agricultural equipment $T$ is decomposed into $n$ subtasks $ST_i$, so $T = \{ST_1, ST_2, \ldots, ST_n\}$.

2. Subtask classification: according to the requirements of processing technology, the corresponding cloud service assemblage is matched for each subtask, where $CSA$ represents the cloud service assemblage which contains $m$ factories $F$.

3. Subtasks matching: after analyzing the manufacturing capability of $m$ factories in a cloud service assemblage, QoS evaluation model is used to match each subtask $ST_i$ to factory $F_{ij}$, where $i$ represents the $i$th subtask, $i = 1, 2, \ldots, n$, and $j$ represents the $j$th factory, $j = 1, 2, \ldots, m$.

4. Subtasks optimization: since there are hundreds of factories, subtasks optimization is a complicated problem, so the optimal manufacturing scheme is realized by optimization algorithm.

2.2. QoS Evaluation Model Analysis. Quality of Service (QoS) evaluation model is often used in various optimization problems [18,19], so QoS evaluation model is adopted for cloud service optimization in this paper. Considering the universality of computation methods, only reliability ($RE$), manufacturing level ($ML$), scale ($SE$), and distance ($DT$) are used as evaluation index to establish the QoS evaluation model.
Reliability (RE). The reliability is mainly determined by the quality of production and the reputation of factories. The factory that has higher reputation will have higher reliability in agricultural equipment manufacturing. In the busy time, agricultural equipment bears heavy production tasks, and it is easy to malfunction. So, the failure rate must be reduced to ensure agricultural production, which put forward a higher request to the reliability of agricultural equipment. In order to gain the trust of farmers, the factory with high reliability must be given priority. This part of the data is given by factories.

Manufacturing Level (ML). The manufacturing level is mainly determined by the ability of the technical staff and the accuracy of the manufacturing workshop. The factory that has higher manufacturing level will produce better agricultural equipment. This part of the data is given by factories.

Scale (SE). Scale of factory can represent the efficiency of manufacturing. The time spent in the manufacturing process will be shorter with the larger scale of factory, and it is the ideal situation of agricultural equipment manufacturing in busy farming. This part of the data is given by factories.

Distance (DT). Distance can represent the cost during transport, which is mainly used to solve the TSP problem in the process of manufacturing. This part of the data is transformed from coordinates that are provided by factories.

Reliability Feedback Update. Massive components are used in the process of manufacturing agricultural equipment, and these components are divided into standard components and nonstandard components. However, nonstandard components should be to processed on-site and the reliability is influenced by raw materials, personnel, and other external factors. In order to ensure the quality of agricultural equipment, the reliability of nonstandard components must be strictly controlled. So, the feedback update strategy is used to update the reliability of each factory in time after each round of manufacturing.

Dynamic Coefficient (P). From the above, the reliability of factory plays a leading role in the QoS evaluation model. For standard components, the reliability is high and does not need too much consideration; for nonstandard components, the reliability changes dramatically, so the factory with high reliability must be given priority. The reliable factories should be selected to further ensure the quality of agricultural equipment by adjusting the proportion of the reliability in the QoS evaluation model. If the fixed coefficient is used to adjust the proportion of reliability, some discontinuous points will appear. Therefore, the results will have strong fluctuations. According to data analysis, a dynamic coefficient method is proposed to solve this problem.

The higher the reliability of factory is, the greater the proportion of reliability is in the evaluation model. It can be considered that there is a certain function between the reliability and the coefficient. After investigation and data fitting, the dynamic coefficient can be calculated by the following equation:

$$ P = \frac{RE}{100} $$

Due to the difference in quantitative unit and range interval of each evaluation index, the data is normalized to improve the accuracy of the evaluation model. The evaluation index can be divided into positive and negative attributes [20]. The positive attributes ($q^+$) include reliability, manufacturing level, and scale, which are calculated by (2). The negative attributes ($q^-$) include distance, which is calculated by (3). The QoS evaluation model can be expressed as (4):

$$ QF = c_1pQRE + c_2(1-p)QML + c_3(1-p)QSE + c_4(1-p)QDT. $$

The combination evaluation index can be regarded as the aggregation of each evaluation index, and the aggregation method is related to the structure of subtasks [21]. In this paper, the manufacturing schemes have sequential structure, so the specific aggregation function is shown in Table 1. The QoS value can be calculated by (5), where $c_1, c_2, c_3, c_4$ are given by experts:

3. Basic Algorithm

At present, artificial intelligence algorithm is widely used in optimization problems and has achieved good results. DAABA is an improved algorithm of the improved artificial ant colony algorithm and artificial bee colony algorithm.

3.1. MAX-MIN Ant System (MMAS). MMAS was proposed by Stutzle, which is an improved algorithm of ACO [22]. The differences between ACO and MMAS are as follows:

(1) MMAS uses an elite strategy to update pheromones. Compared with ACO, MMAS uses the iterative
optimal solution to update the pheromones, which improves the global optimization ability of the algorithm. In order to ensure the validity of pheromone, MMAS updates pheromones according to (6), where $i$ and $j$ are optimal paths:

$$
\begin{align*}
\tau_{i,j}(t + 1) &= (1 - \rho)\tau_{i,j}(t) + \delta\tau_{i,j}, \quad i, j \in m, \\
\delta\tau_{i,j} &= \frac{Q}{FQ_{\text{all}}}, \\
FQ_{\text{all}} &= \sum_{i=1}^{m} \sum_{j=1}^{m} FQ, \quad i, j \in m.
\end{align*}
$$

(2) The pheromone values are limited to $[\tau_{\text{min}}, \tau_{\text{max}}]$, where $\tau_{\text{min}}$ and $\tau_{\text{max}}$ are the lower and upper limits of the pheromone. This method reduces the possibility of falling into the stagnation state and improves the ability of global optimization.

(3) The initial pheromone is set as $\tau_{\text{max}}$ and a volatilization rate is set as $\rho$.

(4) MMAS uses the smoothing mechanism to improve its performance, which can avoid early convergence to the local optimal solution.

The operations of MMAS including parameters initialization, pheromone initialization, heuristic function, state transition rules, fitness function, and pheromone update are shown in Table 2.

The advantages of MMAS are mainly shown in the following two aspects:

(1) The algorithm has the potential of parallelism, which leads to strong robustness and high efficiency.

(2) The algorithm has strong global searching ability in the early stage.

3.2. Artificial Bee Colony (ABC) Algorithm. In 2005, Karaboga proposed the artificial bee colony algorithm [24]. By virtue of strong global search ability, ABC has been successfully applied to the optimization problems. ABC consists of four basic elements: honey sources, employed bees, onlooker bees, and scout bees. These elements are described as follows:

(1) **Honey sources**: each honey source represents a feasible solution to the optimization combination problem, and the fitness of the honey source represents the quality of the feasible solution.

(2) **Employed bees**: the employed bees inform other bees of the honey sources.

(3) **Scout bees**: scout bees randomly search for honey sources near the hive.

(4) **Onlooker bees**: onlooker bees wait in the hive for the honey sources information that the employed bees bring back. According to the quality of the honey sources, onlooker bees follow the appropriate employed bees to mine honey.

The basic operations of ABC including the fitness function of the honey sources, following probability, and detection are shown in Table 3.

The search range is determined at random in ABC, so the ability of global search is enhanced. However, the low accuracy in the early stage and slow convergent speed in later stage are obvious disadvantages.

### 4. Dynamic Artificial Ant-Bee Colony Algorithm Design

Considering the advantages and disadvantages of MMAS and ABC, the DAABA is proposed in this paper. Compared with improved ant colony algorithm and traditional artificial bee colony algorithm, DAABA is improved in three aspects:

(1) Improve the accuracy of search in the early stage. In the first place, DAABA invokes the MMAS to get the better basic solutions. When the MMAS shows stagnation behavior in the later stage, the optimal fusion evaluation strategy is adopted to reduce useless computation.

(2) Enhance the ability of global search. Compared with traditional artificial bee colony algorithm, the one with mutation operation can enhance the ability of global search.

(3) Optimize the capability of search in the later stage. When the feasible solution is near the optimal solution, the iterative adjustment threshold strategy is adopted to increase the population of bee colony. So, the optimal solution can be found more quickly.

#### 4.1. Optimal Fusion Evaluation Strategy

The traditional hybrid algorithm controls the combination point by setting a fixed number of iterations, even though the convergence effect is not ideal. The algorithm has to execute several useless iterations before ABC is invoked, which makes the speed of the hybrid algorithm slow. For the sake of accelerating the speed, the optimal fusion evaluation strategy is adopted. When the stagnation behavior of MMAS appears, ABC is invoked promptly to avoid the waste of computation resources. The specific steps of the optimal fusion evaluation strategy are as follows:

---

**Table 1: QoS aggregate function.**

| Reliability | Manufacturing level | Scale | Distance |
|-------------|---------------------|-------|----------|
| $Q_{RE} = \sum_{i=1}^{n} q_{RE}(s_i)$ | $Q_{ML} = \sum_{i=1}^{n} q_{ML}(s_i)$ | $Q_{SL} = \sum_{i=1}^{n} q_{SL}(s_i)$ | $Q_{FR} = \sum_{i=1}^{n} q_{FR}(s_i)$ |

**Table 2:**
Mathematical Problems in Engineering 5

Table 2: MMAS design methods.

| Basic operation | Method |
|-----------------|--------|
| Parameters initialization | Parameters are randomly generated according to the QoS model |
| Pheromone initialization | All paths are $\tau_{\text{max}}$ |
| Heuristic function | The heuristic function ($\eta$) provides information for $F_i$ to $F_j$. Take $\eta = FQ_{i,j}$, where $FQ_{i,j}$ represents the QoS value between $F_i$ and $F_j$ |
| State transition rules | According to [23], $P_k^{i,j}(t) = \begin{cases} \tau_{ij}(t)f_j^{\beta}(t)/\sum_{k \in \alpha_i} \tau_{ij}(t)f_j^{\beta}(t), & \text{if } j \in \alpha_i, \\ 0, & \text{if } j \notin \alpha_i. \end{cases}$ |
| Fitness function | The fitness function is calculated by (5) |
| Pheromone update | The pheromone update is divided into local update and global update. The calculation method of local update is shown in (6). The calculation method of global update is shown as follows: $\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t), i \in m, j \in m$, where $i$ and $j$ are not optimal paths |

Table 3: ABC design methods.

| Basic operation | Method |
|-----------------|--------|
| Fitness function | The onlooker bees select the honey sources by roulette and following probability. The following probability is calculated according to the following formula: $P(i) = \text{fit}(i)/\sum_{j=1}^{FN} \text{fit}(i)$, where $FN$ represents the number of honey sources |
| Following probability | When the number of minings of the honey source exceeds the limit, a new source is detected according to the following formula: $x(i+1) = x(i) + \text{rand}(x_{\text{max}} - x_{\text{min}})$ |
| Detection | $\delta < \delta_{\text{min}}$, (9) $x(i+1) = (x_{\text{max}} - x_{\text{min}})\text{rand}(1, FN)$. (10) |

4.2. Mutation Artificial Bee Colony Algorithm. When the scout bees detect better honey sources, honey sources are updated. In order to reduce the computation burden, the mutation operation is added to artificial bee colony algorithm to obtain mutation artificial bee colony algorithm, which updates the nonideal honey sources directly. The specific steps for improvement are shown as follows:

(1) The minimum and maximum number of iterations for MMAS are set as $NA_{\text{min}}$ and $NA_{\text{max}}$.

(2) When (7) is satisfied, MMAS is terminated and ABC is invoked:

$$\begin{cases} NA > NA_{\text{min}} \\ NA < NA_{\text{max}} \\ \Delta \text{fit}(t) < \Delta \text{fit}(t+1) \end{cases}$$ (7)

4.3. Iterative Adjustment Threshold Strategy. In the early stage, DAABA utilizes the strong global search ability of MMAS and quickly converges to range of the optimal solution. In the later stage, DAABA makes use of the excellent optimization ability of ABC and searches quickly for the optimal solution. The accuracy of ABC is directly related to the population of bee colony, so the larger the population is, the higher the searching accuracy is. However, the calculation burden will rapidly increase with the large population of bee colony.

In order to improve the accuracy of the DAABA while minimizing the computational complexity, the iterative adjustment strategy is used in this paper. A threshold $NB_j$ is designed to adjust the time of increasing the population. $NB_j$ can be calculated by (11), where $NB_{\text{max}}$ represents the maximum number of iterations and $\phi$ represents the adjustment coefficient.

$$NB_j = \phi \times NB_{\text{max}}, \quad \phi \in (0.7, 0.9).$$ (11)

When the number of iterations in ABC is greater than $NB_j$, the population of bee colony is increased to further find the optimal solution.

4.4. DAABA Operation Process. After analyzing the improved methods of DAABA, the flow chart of DAABA is shown in Figure 2, and the specific steps are shown as follows:

(1) The degree of polymerization for honey sources: the more concentrated the fitness value of the honey sources is and the higher the degree of polymerization is, the computing resource may be wasted due to high-probability stagnation behavior. The degree of polymerization for honey sources can be calculated by the following equation:

$$\delta^2 = \frac{\sum_{i=1}^{FN} (\text{fit}(i) - \text{fit}_{\text{av}})^2}{FN - 1}.$$ (8)

(2) Honey sources update: when (9) is satisfied, the process of mining is not needed. According to mutation operation, honey sources can be directly updated by (10):
Initialization parameters

MMAS to construct paths

Whether to build complete path

Y

Update global pheromone

Whether to meet the best fusion point evaluation strategy according to Eq.(7)

Y

N

Y

Employed bee

Whether the polymerization of honey source is lower than setting

Y

Onlooker bee

Update honey sources

Increase population size

ABC iterations perform $NB_{+1}$

Y

Scout bee

Whether $NB$ is greater than $NB_{max}$

Y

Output optimal solution

MMAS iterations perform $NA_{+1}$

BEGIN

whether $NB$ is greater than $NB_{max}$

Y

Whether $NB$ is greater than $NBJ$

Y

Figure 2: The flow chart of DAABA.
Step 1. Initialize the algorithm parameters $NB_{\min}$, $NB_{\max}$, $NA_{\max}$, $a$, $\beta$, $\phi$, and so on.

Step 2. Construct ant colony path and update pheromone according to the solution of ABC.

Step 3. Determine whether the optimal fusion evaluation strategy is satisfied. If so, terminate MMAS and invoke ABC; otherwise, go to Step 2.

Step 4. ABC enters the stage of employed bees and calculates the fitness value according to the information of honey sources.

Step 5. Judge the degree of polymerization for honey sources according to fitness value. If the degree of polymerization is higher than expectation, update the honey sources and go to Step 4; otherwise, go to the next step.

Step 6. According to the fitness value of each honey source, onlooker bees select honey sources by the roulette.

Step 7. Scout bees randomly search for new honey sources.

Step 8. Determine whether the iteration of ABC $NB$ is greater than adjustment threshold $NB_j$. If so, increase the population of bee colony; otherwise, continue to operate with the original population.

Step 9. Determine whether the iteration of ABC $NB$ is greater than maximum iterations of ABC $NB_{\max}$. If so, output the optimal solution; otherwise, go to Step 8.

4.5. Convergence Analysis of DAABA. In DAABA, MMAS is only embedded into the ABC, so the structural integrity of the ABC has not been destroyed. Therefore, the convergence of DAABA and ABC is consistent. According to [25], MMAS is convergent. So, DAABA is verified to be convergent according to [26] and the following equation:

$$|\psi + \phi| < 1,$$

$$|\psi - \phi| < 1.$$  \hspace{1cm} (12)

4.6. Time Complexity Analysis of DAABA. Time complexity is an important indication for running efficiency of the algorithm. In DAABA, $k$ represents the population of ant and bee colony, $n$ represents the number of factories in the cloud service assemblage, and $m$ represents the number of subtasks. The time complexity of MMAS can be expressed as $T1(n) = O(k(nm + m) + 1)$, and the time complexity of ABC can be expressed as $T2(n) = O(k(nm + m^2) + 1)$, so the time complexity of DAABA is expressed as $T(n) = O(2km + km^2 + km + 2)$.

5. Simulation and Experiment

5.1. Simulation. In order to ensure that agricultural equipment manufacturing can be finished in time, simulation experiments are needed to verify the reliability of DAABA. The specific steps of simulation are shown as follows:

1. The agricultural equipment manufacturing tasks are numbered from 1 to 6. Each task is divided into 10 subtasks, which includes the processing technics of die forging, extrusion, rolling, drawing, cutting, fusion welding, water jet cutting, plasma cutting, milling, and shearing.

2. The factories are numbered from 1 to 40 of each cloud service assemblage, and the manufacturing information is processed according to the evaluation index.

3. DAABA is invoked and the initial parameters are shown in Table 4. The experimental environment is PC, CPU Intel core i5-7300 2.50GHz, 16GB, Windows 10, MATLAB 2015b.

4. The optimization result of cloud service in agricultural equipment manufacturing is shown in Figure 3. The result shows that DAABA tends to converge after 200 iterations, which proves the convergence of DAABA again.

5. The multiple agricultural equipment manufacturing tasks are carried out simultaneously through task assignment of cloud service pool. The result of multiple manufacturing tasks is shown in Figure 4, where the horizontal and vertical axes represent the virtual plane coordinates, each point represents a factory, and each color loop represents a complete cloud manufacturing scheme of agricultural equipment.

5.2. Comparison Experiment

5.2.1. Accuracy Experiment. In order to verify the accuracy of DAABA for cloud service optimization, the experiments based on DAABA, GA, MMAS, and ABC are carried out. The control parameters for each algorithm are shown in Table 4. All algorithms have been tested for 9 large-scale problems, and the scale of the test is shown in Table 5. Each algorithm is repeated 20 times to reduce the error, and the optimization results are shown in Figure 5, where the task size is expressed as $T(n,m)$, $m$ is the number of subtasks, and $n$ is the number of factories in cloud service assemblage of each subtask.

5.2.2. Speed Experiment. In order to verify the speed of DAABA for cloud service optimization, the experiments on DAABA and ABC are carried out. The average time consumption for each algorithm is shown in Figure 6.

5.3. Results Analysis. Figure 5 shows that the accuracy of DAABA is better than those of the others. As the data increases, the advantages of DAABA and ABC are more obvious. This is because ABC is more accurate in solving high-dimensional problems. Furthermore, the honey sources are generated according to the results of MMAS in the...
early stage and the colony size is increased by using iterative adjustment threshold strategy in the later stage, so the accuracy of DAABA is higher than that of ABC.

Figure 6 shows that the speed of DAABA is faster than that of ABC. This is because optimal fusion evaluation strategy is introduced to reduce the useless iterations.
In summary, the performance of DAABA is superior to those of the others, and DAABA is more suitable to solve large-scale problem in cloud service optimization of agricultural equipment.

6. Conclusion

Massive nonstandard components are used in the manufacturing process of agricultural equipment and their quality is difficult to be quantified, which limits the efficiency and accuracy of cloud service in agricultural equipment manufacturing. In response to this problem, a cloud service optimization method is proposed in this paper, in which the reliability feedback update strategy and the dynamic coefficient strategy are presented to adjust the proportion of the reliability in the evaluation model and improve the quality of agricultural equipment. Then a dynamic artificial ant-bee colony algorithm (DAABA) is proposed to enhance the accuracy and efficiency of cloud manufacturing of agricultural equipment. In DAABA, the optimal fusion evaluation strategy is adopted to invoke ABC in time, which can reduce...
the unnecessary iteration, shorten the computing time, and improve the running speed of cloud service; the mutation operation is added to the ABC to update the honey sources directly, so the ability of optimization is maximized; and the iterative adjustment threshold strategy is used, which can increase the population of bee colony to improve the accuracy of cloud service while minimizing the computational complexity in later stage. Finally, the convergence of DAABA is verified through theoretical derivation and simulation. Then experiments based on DAABA, GA, MMAS, and ABC are carried out to verify the performance of DAABA. The results show that the accuracy and efficiency of DAABA are superior to those of the others on cloud service optimization in agricultural equipment manufacturing.

Data Availability

The .mat data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by China Postdoctoral Foundation (Grant no. 2019M662410) and the National Defense Basic Scientific Research Program of China (Grant no. JCKY2016204A502).

References

[1] G. Geng and K. Li, “Research and application of digital design and manufacturing technology of agricultural machinery,” Agro Food Industry Hi-Tech, vol. 28, no. 1, pp. 2891–2894, 2017.
[2] G. Ozogul, “Effects of changing concepts and conditions of agricultural production to agricultural machinery manufacturing sector in globalizing world,” Journal of Food Agriculture and Environment, vol. 10, no. 3-4, pp. 456–462, 2012.
[3] B. Ben, R. Ihab, and L. Wang, “Cloud manufacturing: current trends and future implementations,” Journal of Manufacturing Science & Engineering, vol. 137, no. 4, Article ID 044001, 2015.
[4] Y. Cao, S. Wang, L. Kang, C. Li, and L. Guo, “Study on machining service modes and resource selection strategies in cloud manufacturing,” The International Journal of Advanced Manufacturing Technology, vol. 81, no. 1–4, pp. 597–613, 2015.
[5] Y.-L. Luo, L. Zhang, F. Tao et al., “A modeling and description method of multidimensional information for manufacturing capability in cloud manufacturing system,” The International Journal of Advanced Manufacturing Technology, vol. 69, no. 5–8, pp. 961–975, 2013.
[6] M. Czygan, L. Kowalik, and A. Socala, “Improving TSP tours using dynamic programming over tree decompositions,” European Symposium on Algorithms, vol. 15, no. 4, pp. 1–19, 2019.
[7] J. Ding, Y. Wang, S. Zhang, W. Zhang, and Z. Xiong, “Robust and stable multi-task manufacturing scheduling with uncertainties using a two-stage extended genetic algorithm,” Enterprise Information Systems, vol. 13, no. 10, pp. 1442–1470, 2019.
[8] S.-L. Wang, L. Guo, L. Kang et al., “Research on selection strategy of machining equipment in cloud manufacturing,” The International Journal of Advanced Manufacturing Technology, vol. 71, no. 9–12, pp. 1549–1553, 2014.
[9] S. T. Al-Shihabi and M. M. Aldurgam, “A max-min ant system for the finance-based scheduling problem,” Computers & Industrial Engineering, vol. 110, pp. 264–276, 2017.
[10] W.-F. Gao and S.-Y. Liu, “A modified artificial bee colony algorithm,” Computers & Operations Research, vol. 39, no. 3, pp. 687–697, 2012.
[11] J.-J. Zhou and X. Yao, “A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition,” The International Journal of Advanced Manufacturing Technology, vol. 88, no. 9–12, pp. 3371–3387, 2017.
[12] I. Aydin, “A new approach based on firefly algorithm for vision-based railway overhead inspection system,” Measurement, vol. 74, pp. 43–55, 2015.
[13] A. Kumar and S. Bawa, “Generalized ant colony optimizer: swarm-based meta-heuristic algorithm for cloud services execution,” Computing, vol. 101, no. 10, pp. 1609–1632, 2019.
[14] E. J. Gholni, A. Rahmani, and N. N. Qader, “Service load balancing, scheduling, and logistics optimization in cloud manufacturing by using genetic algorithm,” Concurrency and Computation: Practice and Experience, vol. 31, no. 20, Article ID e5329, 2019.
[15] K. Duan, S. Fong, S. Siu, W. Song, and S. Guan, “Adaptive incremental genetic algorithm for task scheduling in cloud environments,” Symmetry, vol. 10, no. 5, 168 pages, 2018.
[16] Y.-F. Yang, B. Yang, S. Wang et al., “A dynamic ant-colony genetic algorithm for cloud service composition optimization,” The International Journal of Advanced Manufacturing Technology, vol. 102, no. 1–4, pp. 355–368, 2019.
[17] J.-J. Zhou and X. Yao, “Hybrid teaching-learning-based optimization of correlation-aware service composition in cloud manufacturing,” The International Journal of Advanced Manufacturing Technology, vol. 91, no. 9–12, pp. 3515–3533, 2017.
[18] S. Garg, K. Modi, and S. Chaudhary, “A QoS-aware approach for runtime discovery, selection and composition of semantic web services,” International Journal of Web Information Systems, vol. 12, no. 2, pp. 177–200, 2016.
[19] Z.-J. Wang, Z. Liu, X. Zhou, and Y. Lou, “An approach for composite web service selection based on DQGQoS,” The International Journal of Advanced Manufacturing Technology, vol. 56, no. 9–12, pp. 1167–1179, 2011.
[20] Q. Wu, F. Ishikawa, Q. Zhu, and D.-H. Shin, “QoS-aware multigranularity service composition: modeling and optimization,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46, no. 11, pp. 1565–1577, 2016.
[21] L.-Z. Zeng, B. Benatallah, A. H. H. Ngu et al., “QoS-aware middleware for web services composition,” IEEE Transactions on Software Engineering, vol. 30, no. 5, pp. 311–327, 2004.
[22] T. Stützle and H. Hoos, “Ant system,” Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000.
[23] Y. Cao, S. Wang, L. Kang, and Y. Gao, “A TQCS-based service selection and scheduling strategy in cloud manufacturing,” The International Journal of Advanced Manufacturing Technology, vol. 82, no. 1–4, pp. 235–251, 2016.
[24] D. Karaboga, “Artificial bee colony algorithm,” Scholarpedia, vol. 5, no. 3, p. 6915, 2005.

[25] T. Stutzle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," IEEE Transactions on Evolutionary Computation, vol. 6, no. 4, pp. 358–365, 2002.

[26] J. C. Bansal, A. Gopal, and A. K. Nagar, "Analysing convergence, consistency, and trajectory of artificial bee colony algorithm," IEEE Access, vol. 6, pp. 73593–73602, 2018.