Dynamic Density Clustering Ant Colony Algorithm With Filtering Recommendation Backtracking Mechanism

JIN YU1, XIAOMING YOU2, AND SHENG LIU2

1 College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
2 School of Management, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Xiaoming You (yxm6301@163.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61673258 and Grant 61075115, and in part by the Shanghai Natural Science Foundation under Grant 19ZR1421600.

ABSTRACT Ant colony system has a good performance in solving Traveling Salesman Problem (TSP), but it tends to fall into local optimum and is deficient in convergence speed. To address this problem, a dynamic density clustering ant colony algorithm with a filtering recommendation backtracking mechanism is proposed (DBACS). Firstly, a dynamic density clustering strategy is proposed to accelerate the convergence speed of the algorithm and improve the quality of the solution. Under this strategy, the search radius is expanded dynamically to merge adjacent classes, so as to form the differential pheromone distribution. The splicing paths between each class are adjusted through the ant colony algorithm to achieve better performance. Secondly, a recommendation backtracking mechanism based on collaborative filtering is proposed to increase the diversity of the population, thus helping the algorithm jump out of the local optimum. With the help of the collaborative filtering algorithm, some dense data points are recommended for pheromone dynamic backtracking, which can not only help algorithm jump out of the local optimum, but also help the algorithm accelerate convergence. Simulation results show that the improved algorithm can obtain a better solution and higher stability. Especially in solving large-scale TSP, the accuracy of the solution is significantly improved.

INDEX TERMS Ant colony algorithm, dynamic clustering, collaborative filtering recommendation, pheromone dynamic backtracking, TSP.

I. INTRODUCTION

Traveling salesman problem (TSP) is a classical combinatorial optimization problem. The content of TSP is to find the shortest path through all cities on the premise that each city is passed only once. Many algorithms can solve TSP, such as Ant Colony Optimization (ACO) [1], Genetic Algorithm (GA) [2], and Particle Swarm Optimization (PSO) [3]. What’s more, some hybrid algorithms show very promising results for TSP, such as the improved Immune Algorithm [4] and the improved Clonal Selection Algorithms [5], [6].

Ant Colony Optimization (ACO) [7], [8] was first proposed in 1996 by Marco Dorigo with reference to the ant foraging mechanism to solve distributed optimization problems. Nowadays, improved ant colony algorithms are used to realize path planning of mobile robots [9], predicting RNA folding path [10], agricultural soil survey [11], data mining [12], etc. Although ant colony algorithm has strong advantages in solving TSP, there are still some deficiencies. When the algorithm solves large-scale TSP, the convergence speed of the algorithm decreases, and it is easy to fall into local optimum. Later, Dorigo combined local pheromone update with global pheromone update and put forward the Min-Max Ant System (MMAS) [14] to prevent the algorithm from falling into local optimum due to too high local pheromone. These traditional ant algorithms have good global search capability and good performance for
small-scale TSP, but for medium-scale and large-scale TSP, there are still problems such as slow convergence speed and poor accuracy.

In order to solve the above problems, many experts and scholars have improved the ant colony algorithm from different fields and aspects.

Saenphon et al. [15] proposed a fast-reverse-gradient-search strategy to enhance the ability of the algorithm to find solutions. Zhang et al. [16] introduced the replication operation of bacteria foraging, which improves the speed of finding solutions by fusing the elimination mechanism. Starzec et al. [17] proposed a parallel ant colony optimization algorithm, which uses high-performance computing infrastructure to make the algorithm run effectively on a cluster of 30 computing nodes. The designed system could run thousands of ants. Although they have accelerated the convergence speed of the algorithm, the diversity of the algorithm is still insufficient.

Gaifang et al. [18] combined ant colony algorithm with genetic algorithm, and used the nearest-neighbor-selection strategy to get a better solution. Meng et al. [19] proposed directional pheromones and exploration factors to improve the quality of the solution. Mahi Ö. K et al. [20] firstly used particle swarm optimization algorithm to optimize the parameters of ant colony algorithm, and then used 3-opt to increase the diversity of the algorithm. Although they have improved the diversity of algorithms, the convergence speed of the algorithm is insufficient.

Pang et al. [21] used the idea of clustering to transform large-scale TSP into small-scale TSP, which improves the convergence speed of the algorithm. Yang [22] used an orthogonal method to cluster targets, which improves the efficiency of the algorithm. Du et al. [23] classified data and ants, and used different search strategies to find solutions, thus increasing the diversity of the solutions. Pang et al. [24] used clustering to generate the correlation matrix, which improves the quality of the solution. These improvements have greatly improved the diversity and convergence speed of the algorithm, but they still have the problems of slow convergence speed and poor accuracy in solving large-scale TSP. The root cause is that there are too many paths for ant colony to choose due to the large TSP data size, which increases the difficulty of finding the high precision solution. At the same time, the high concentration of pheromone also reduces the diversity of the algorithm in the later stage.

The main purpose of this paper is to improve the convergence speed of the algorithm and the quality of the solution when dealing with large-scale TSP. A dynamic density clustering ant colony algorithm with a filtering recommendation backtracking mechanism is proposed. A clustering algorithm and a collaborative filtering recommendation algorithm are adopted to optimize the ant colony algorithm. The improved ant colony algorithm can better help solve the problems encountered in the actual production process, such as improving the production efficiency of the multi-bridge machining system (MBMS) [25]. The main contributions of this paper are as follows:

1. A dynamic density clustering strategy is proposed to improve the poor accuracy of the solution and the slow convergence speed. The strategy includes two parts: primary clustering and dynamic clustering. During the primary clustering, TSP will be classified by Density-Based Spatial Clustering of Applications with Noise (DBSCN). Because the scale of each new TSP generated by clustering is small, ant colony algorithm can obtain good local solutions. Dynamic clustering can merge adjacent classes on the basis of primary clustering and finally form a large class.

2. A recommendation backtracking mechanism based on collaborative filtering is proposed to improve the diversity of the population. Dense data points are recommended by an image processing algorithm for dynamic pheromones backtracking. The pheromone concentration on dense paths will dynamically go back to a certain state which the algorithm has got before.

The paper is organized as follows: Section II briefly introduces the ACS and DBSCN. Section III introduces the improved strategies in DBACS, including the dynamic density clustering strategy and the recommendation backtracking mechanism based on collaborative filtering. Section VI verifies the effectiveness of the relevant strategies and compares DBACS with the traditional ant colony algorithm and the latest improved algorithms. Section V summarizes the paper and proposes the future work.

II. RELATED ALGORITHM

A. ANT COLONY SYSTEM (ACS) FOR TSP

1) PATH SELECTION RULE

In the ACS, each ant in the city \( i \) will choose its next city \( j \) according to the pseudo-random ratio, referring to formula (1).

\[
    j = \begin{cases} 
    \arg \max \{ \tau_{ij} [\eta_{ij}]^\beta \}, & q \leq q_0 \\
    J, & q > q_0 
    \end{cases} 
\]

(1)

where \( q \) is a random variable, which is evenly distributed in the \([0,1]\); The value range of \( q_0 \) is \([0,1]\); \( j \) is a variable obtained according to formula (2). \( \beta \) is the expectation heuristic factor. \( \eta_{ij} \) represents the reciprocal of the distance between cities \( i \) and \( j \), and its formula is (3). \( \tau_{ij} \) is the pheromone concentration between city \( i \) and \( j \). \( J \) is equal to Eq. (2).

\[
P^k_{ij}(t) = \begin{cases} 
\frac{[\tau_{ij}(t)^\alpha][\eta_{ij}(t)^\beta]}{\sum_{s \in J(t)}[\tau_{is}(t)^\alpha][\eta_{is}(t)^\beta]}, & j \in N^k_i \\
0, & j \notin N^k_i 
\end{cases} 
\]

(2)

where \( \alpha \) and \( \beta \) are information heuristic factor and expectation heuristic factor respectively; \( N^k_i \) is the city set that ants can reach directly and have not visited before; \( \eta_{ij} \) represents the reciprocal of the distance between cities \( i \) and \( j \), and its formula (3) is as follows:

\[
\eta_{ij} = \frac{1}{d_{ij}} 
\]

(3)
isolated points. Isolated points can be classified separately. The method can be automatically determined. Classes of any shape and size can be divided, and the method is not influenced by isolated points. Isolated points can be classified separately.

2) PHEROMONE UPDATE
Local pheromone update: the ant in the city $i$ updates the pheromone on the current path immediately after selecting the next city $j$, and its formula (4) is as follows:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0$$

where $\rho$ is the pheromone volatilization coefficient which is in $(0, 1)$; $\tau_0$ is the initial value of the path pheromone between the cities $i$ and $j$.

Global pheromone update: After the ants in the current iteration have finished searching for the solutions, the algorithm updates the global pheromone on the shortest path to speed up the convergence of the algorithm. The formula (5) is as follows:

$$\tau_{ij} = (1 - \xi)\tau_{ij} + \xi \Delta \tau_{ij}^*$$

$$\Delta \tau_{ij}^* = \frac{1}{C^*}$$

where $\xi$ is the global pheromone volatilization coefficient, and $C^*$ is the shortest path length found during this iteration. $\Delta \tau_{ij}^*$ is the pheromone added to the globally optimal path during this iteration. The calculation formula is (6).

B. DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE (DBSCAN)
The main capability of the algorithm is to actively find the high-density points, and then connect the nearby high-density points to generate a cluster. The concrete implementation process of the algorithm is to set $\varepsilon$ as the radius and then draw a circle with each data point as the center of the circle. The algorithm will calculate the number of points in the circle, which is the density value of the point. Then, a density threshold $MinPts$ is selected. For example, the center point with the number of circles less than $MinPts$ is a low-density point, while the center point with the number of circles greater than or equal to $MinPts$ is a high-density point. If there is a high-density point within the circle of another high-density point, the algorithm will connect the two points. If there is a low-density point within the circle of high-density points, it is connected to the nearest high-density point, which is called the boundary point. In this way, all the points that can be connected form a cluster, and the points that are not in any high-density circle are isolated. The formula is shown in (7) and (8).

$$N(x_j) = \{x_i \in D | \text{distance}(x_i, x_j) \leq \varepsilon\}$$

$$|N(x_j)| \geq MinPts$$

where $D$ is the sample data set, $N(x_j)$ is the number of sub-sample sets; $\varepsilon$ is the set distance, and $x_j$ is the core object; $MinPts$ is the density threshold.

The method has the advantages that the number of classes can be automatically determined. Classes of any shape and size can be divided, and the method is not influenced by isolated points. Isolated points can be classified separately.

III. ALGORITHM IMPROVEMENT (DBACS)
Ant colony algorithm has many problems in the solution of TSP, especially when solving large-scale TSP, such as slow convergence speed and easy to fall into local optimum. Therefore, this paper proposes a dynamic density clustering strategy and a recommendation backtracking mechanism based on collaborative filtering.

A. DYNAMIC DENSITY CLUSTERING STRATEGY
1. DBSCAN classifies the current data set based on density. Firstly, the current data set is clustered for the first time with a suitable radius, in which isolated points form a class of their own. Fig. 1 is a result of the primary clustering of data set pr226.

After clustering is completed, each small class is treated as a new TSP to be solved, and the traditional ant colony algorithm has the advantage of high precision in solving the small-scale TSP, so that the algorithm can converge in a short time, as shown in Fig. 2.

Through this method, the algorithm can quickly construct the initial differential pheromone distribution of the path in the early stage, which provides preparation for the connection of the following connecting-ants. This can greatly increase the convergence speed of the algorithm while ensuring the accuracy of the algorithm.

2. After completing the above work, the algorithm enters the process of dynamic clustering. After the primary clustering reach convergence, DBSCAN begins to expand the search radius and re-cluster the original data set. The newly generated classes are formed based on the original classes and adjacent classes. As shown in Fig. 3, it is a result of secondary clustering.
After clustering again, the number of classes decreases, and the size of each class increases, forming a new TSP. Then, ant colony algorithm is used again to find the shortest path. Because the newly formed classes are based on the original ones, most of the paths have high concentrations of pheromones. The ant colony is induced by high-concentration pheromones on these paths so only a small number of paths need to be re-selected. This way can greatly reduce the difficulty of finding the solution of the algorithm. After the convergence of the algorithm is completed, the above operations will continue to be repeated. The number of classes will be reduced by expanding the search range again until a class is obtained. As shown in Fig. 4, pr226 eventually becomes a large class.

Due to the high accuracy of local solutions obtained by the primary clustering, the quality of the globally optimal solution obtained by dynamic clustering is improved. Through the aggregation of small classes into large classes, the relatively correct distribution of pheromones is realized to induce ants to make path selection, so as to avoid blindness in finding solutions and ensure accuracy.

B. RECOMMENDATION BACKTRACKING MECHANISM BASED ON COLLABORATIVE FILTERING

Ant colony algorithm is prone to stagnation when solving TSP because of its positive feedback characteristic. Especially when dealing with large-scale TSP, the algorithm is easy to fall into local optimum in the early stage, thus losing the ability to find solutions. The traditional optimization method 3-opt takes too long and has a limited effect in solving large-scale TSP data sets. It is difficult to help the algorithm jump out of the local optimum. The root cause is that the increasing pheromones affect ants’ choices of routes, which leads to a decrease in the diversity of the algorithm.

Through experiments, it can be found that ant colony algorithm has poor ability to find solutions mainly in places with dense data points. It is prone to cross-paths and suboptimal-paths, which reduces the diversity of the algorithm and results in poor solutions or stagnation of the algorithm.

To solve the above problems, a recommendation backtracking mechanism based on collaborative filtering is proposed in combination with the image processing algorithm (DBSCAN) to increase the diversity of the population. The concentration of pheromones in dense paths will dynamically go back to a certain state which the algorithm has got before.

1. The collaborative filtering recommendation algorithm is a common computer algorithm. This strategy uses a clustering algorithm to carry out collaborative filtering. First, a density threshold $\text{MinPts}$ is set, and then DBSCAN is used to identify the current data set. The following Fig. 5 is the result of the density identification of data set eil51. With the help of the collaborative filtering strategy, the data set eil51 is filtered out of some data points, leaving only 8 small classes.

2. After the collaborative filtering is completed, the strategy enters the process of recommendation backtracking. From the previous analysis, the dense place is the key to affect the accuracy of the solution. The results of collaborative filtering are the points which are recommended by the algorithm for backtracking. Some parts from the recommended data points will be backdated randomly, whereas the points not in the recommended list will remain the same. The pheromone of the point where the backtracking is performed will automatically go back to a certain state which has got before.
In Fig. 6, (a) is the pheromone concentration of the data set eil51 in a certain state, and (b) is the state of pheromone concentration after the recommended backtracking mechanism based on collaborative filtering. It can be found that eil51 has undergone pheromone backtracking at some dense locations, which weakens the pheromone concentration.

3. The algorithm will dynamically change the density threshold according to the running situation, which leads to different results after each collaborative filtering. Different recommended data points will lead to different results, thus greatly increasing the possibility of jumping out of the local optimum. This strategy can effectively increase the diversity of the algorithm and reduce the disadvantages brought by the positive feedback of ant colony algorithm in the later period. Its advantages are mainly manifested in two aspects. First, dynamic small-scale backtracking will not affect the convergence speed of the algorithm, especially in solving large-scale TSP. Local pheromone backtracking gives ants the opportunities to choose again, which can effectively help the algorithm get rid of the local optimal solution currently trapped. Second, the strategy combines the advantages of the collaborative filtering recommendation algorithm to make the backtracking mechanism more targeted. It can be found that the algorithm is most likely to discover the locations of the poor paths in the current solution searching process. It can dynamically backdate pheromones in some dense places, greatly improving the accuracy of backtracking, which increases the possibility of the algorithm to jump out of the local optimum.

C. ALGORITHM STEPS
Step 1: Initialize parameters and pheromone matrices. Calculate the distance between cities.
Step 2: Ants are divided into clustering-ants and connecting-ants.
Step 3: The current data set is clustered by DBSCAN, according to Eq. (7) and Eq. (8).
Step 4: Clustering-ants are placed randomly in different cities in a certain class, and connecting-ants are placed randomly in the entire data set.
Step 5: Clustering-ants search for the shortest path in each class, and update the local and global pheromone in each class according to Eq. (4) and Eq. (5).
Step 6: Connecting-ants connect classes to form a global optimal path according to pheromones left by clustering-ants and update the local and global pheromone according to Eq. (4) and Eq. (5).
Step 7: Expand the search radius of DBSCAN and re-cluster the original data until the algorithm gets a cluster.
Step 8: The recommendation backtracking mechanism based on collaborative filtering is used to increase the diversity of the population and help the algorithm jump out of the local optimum when the algorithm stagnates.
Step 9: The number of iterations increases and the procedure goes back to step 4.
Step 10: When the maximum number of iterations is reached, the algorithm outputs the best global solution.

D. TIME COMPLEXITY ANALYSIS
From the flow chart of the algorithm in Table 1 and Fig. 7 above, it can be seen that the time complexity of the algorithm DBACS in this paper is \( o(c^a(n + n^*)w) \), where \( c \) is the number of cities; \( n \) is the number of clustering-ant; \( n^* \) is the number of connecting-ant \( (n + n^* = H) \), and \( w \) is the maximum number of iterations. Therefore, the time complexity of DBACS is \( o(c^aH^aw) \). It shows that the improved algorithm in this paper does not increase the complexity of the algorithm.

IV. SIMULATION EXPERIMENT AND RESULT ANALYSIS
A. SIMULATION ENVIRONMENT AND PARAMETER SETTING
In order to verify the effectiveness of the improved algorithm, the experimental environment of this paper is the WIN10 operating system. Matlab2019a is used to simulate different sizes of TSP in TSPLIB [26]. Moreover, we use the Taguchi’s experimental design method to set parameters [27]. Each combination of the various parameters was tested 15 times, and the average value was taken for analysis. We take eil51 as an example to determine the parameter
Based on the above experiments, it can be concluded that: in ACS, the best scheme of parameters is that $\alpha$ is equal to 1, $\beta$ is equal to 4, and $\rho$ is equal to 0.1, and $\xi$ is equal to 0.3, and $q_0$ is equal to 0.8; in MMAS, the best scheme of parameters is that $\alpha$ is equal to 1, $\beta$ is equal to 1, and $\xi$ is equal to 0.2, and $\xi_0$ is equal to 0.7.

**B. EFFECTIVENESS ANALYSIS OF IMPROVEMENT STRATEGIES**

In this paper, the dynamic density clustering strategy and the recommendation backtracking mechanism based on collaborative filtering are used to optimize the traditional ant colony algorithm. In order to verify the effectiveness of the two strategies, pr107 and pr152 are selected for comparative tests. The maximum number of iterations is 2000, and the three algorithms are run 15 times respectively. DBACS is the ant colony algorithm optimized in this paper. DBACS-1 is the improved algorithm without the dynamic clustering strategy. DBACS-2 is the improved algorithm without the recommendation backtracking mechanism based on collaborative filtering. ACS is the traditional ant colony algorithm.
and MMAS is the traditional maximum and minimum ant algorithm. The optimal solution (Best), the worst solution (Worst), the average solution (Mean), the error rate (Er), and the iteration times of the optimal solution (Convergence) are summarized in Table 8.

Analysis of Table 8, Fig. 8 and Fig. 9 shows that in the data set pr107, the three improved algorithms DBACS, DBACS-1 and DBACS-2 all find the optimal solution 44303, whereas the traditional ant colony algorithms ACS and MMAS do not find the optimal solution. In the data set pr152, DBACS and DBACS-2 find the optimal solution 73682, whereas DBACS-1 fails to find the optimal solution, but its accuracy is higher than that of ACS and MMAS. It is observed that the dynamic density clustering strategy and the recommendation backtracking mechanism based on collaborative filtering play a significant role in improving the accuracy and convergence speed of the ant colony algorithm. As can be seen from Fig. 10 and 11, the improved ant colony algorithm with recommendation backtracking mechanism based on collaborative filtering has better diversity than ACS. In the two tests, the DBACS integrated with the two strategies has better performance than DBACS-1 and DBACS-2. The diversity of the algorithm is improved and the optimal solution is found. It shows that the DBACS with two strategies has improved the convergence speed and accuracy.

C. COMPARATIVE ANALYSIS OF IMPROVED ALGORITHM AND TRADITIONAL ALGORITHM

In order to show the improvement of the algorithm DBACS in this paper, some TSP data in TSPLIB are selected for experimental analysis. The results are also compared with traditional ACS, MMAS. In addition, in order to make the comparison more convincing, we select the improved ant colony algorithm in the last two years for comparison, including PACO-3Opt (2018) [28], TREEACS (2019) [29], JCACO (2019) [30], and IFACO (2019) [31]. Our algorithms are respectively carried out 15 times, with the maximum number of iterations for each experiment being 2000. The test platform we used was a laptop computer, and the operating system was WIN10. The test environment was MATLAB2019, the CPU we used was intel core i7-4710mq, and the size of the memory was 8G. The optimal solution (Best), the worst

### Table 6. Orthogonal test scheme and test results of MMAS.

| No. | \( \alpha \) | \( \beta \) | \( \xi \) | \( q_e \) | result |
|-----|--------|--------|--------|--------|--------|
| 1   | 1      | 1      | 0.1    | 0.6    | 430.40 |
| 2   | 1      | 2      | 0.2    | 0.7    | 435.00 |
| 3   | 1      | 3      | 0.3    | 0.8    | 432.00 |
| 4   | 1      | 4      | 0.4    | 0.9    | 432.66 |
| 5   | 2      | 1      | 0.2    | 0.8    | 441.00 |
| 6   | 2      | 2      | 0.1    | 0.9    | 435.13 |
| 7   | 2      | 3      | 0.4    | 0.6    | 434.46 |
| 8   | 2      | 4      | 0.3    | 0.7    | 436.40 |
| 9   | 3      | 1      | 0.3    | 0.9    | 432.73 |
| 10  | 3      | 2      | 0.4    | 0.8    | 435.86 |
| 11  | 3      | 3      | 0.1    | 0.7    | 432.00 |
| 12  | 3      | 4      | 0.2    | 0.6    | 440.20 |
| 13  | 4      | 1      | 0.4    | 0.7    | 435.06 |
| 14  | 4      | 2      | 0.3    | 0.6    | 433.06 |
| 15  | 4      | 3      | 0.2    | 0.9    | 430.40 |
| 16  | 4      | 4      | 0.1    | 0.8    | 435.00 |

### Table 7. Analysis of test results of MMAS.

| \( T \) | \( \alpha \) | \( \beta \) | \( \xi \) | \( q_e \) |
|--------|--------|--------|--------|--------|
| \( T_1 \) | 434.183 | 434.697 | 434.695 | 439.348 |
| \( T_2 \) | 435.197 | 435.065 | 433.480 | 432.715 |
| \( T_3 \) | 434.862 | 434.798 | 435.048 | 432.965 |
| \( T_4 \) | 435.080 | 434.762 | 436.100 | 434.295 |

### Table 8. Analysis of test results of MMAS.
TABLE 8. Strategy comparison of various algorithms.

| TSP Instances | Algorithm | Best    | Worst   | Mean    | Er      | Convergence |
|---------------|-----------|---------|---------|---------|---------|-------------|
| Pr107         | DBACS     | 44303   | 44624   | 44437   | 0.00    | 85          |
|               | DBACS-I   | 44303   | 44753   | 44538   | 0.00    | 308         |
|               | DBACS-2   | 44303   | 44621   | 44454   | 0.00    | 437         |
|               | ACS       | 44438   | 44830   | 44564   | 0.30    | 1874        |
|               | MMAS      | 44607   | 47217   | 45603   | 0.69    | 1199        |
| Pr152         | DBACS     | 73682   | 74599   | 74015   | 0.00    | 631         |
|               | DBACS-I   | 73818   | 74298   | 74045   | 0.18    | 309         |
|               | DBACS-2   | 73682   | 74960   | 74041   | 0.00    | 1575        |
|               | ACS       | 74139   | 75425   | 74576   | 0.62    | 1379        |
|               | MMAS      | 74404   | 77334   | 75920   | 0.98    | 1545        |

FIGURE 10. Pr107 diversity comparison.

FIGURE 11. Pr152 diversity comparison.

The Wilcoxon rank-sum test is implemented to detect whether the difference between every pair of algorithms is significant. We used DBACS to perform rank-sum tests respectively with ACS and MMAS. The results of the Wilcoxon rank-sum test are summarized in column $P$. If the minimum error rate is expressed by equation (9).

\[ Er = \frac{L_{ACO} - L_{min}}{L_{min}} \times 100\% \]  

$L_{ACO}$ is the optimal solution obtained by the algorithm and $L_{min}$ is the standard optimal solution of the TSP data set.
As can be seen from Table 9, the DBACS in this paper performs well in solving the small-scale TSP. The algorithm can find the standard optimal solution and the convergence speed is faster. From (a) to (f) in Fig. 12, it serves to show that the algorithm has already reached convergence in the early stage and has high accuracy.
In dealing with large-scale TSP, DBACS has obvious advantages. The first advantage is the fast convergence speed. Compared with the traditional Ant Colony System (ACS) and Maximum and Minimum Ant System (MMAS), DBACS can complete the initial distribution of pheromone in a short time and construct a relatively correct path. Relying on the advantages of dynamic clustering, ant colony algorithm can be used in a small range to obtain more accurate solutions, and pheromone induction is helpful to accelerate the convergence of the algorithm. However, the other two algorithms are blind in finding the solution in the early stage and need to spend a lot of time constructing a relatively good initial path. From (f) to (n) in Fig. 12, the ability of the traditional algorithm to find the solution begins to decline when solving the large-scale TSP, and it is easy to fall into local optimum. It can be noticed from Fig. 12 that the three algorithms have fallen into a local optimum in the later period. DBACS with a recommendation backtracking mechanism based on collaborative filtering plays a role in improving the accuracy of the solution. A collaborative filtering recommendation algorithm is used to identify the places where are relatively dense, and targeted dynamic backtracking is implemented to improve the possibility that the algorithm jumps out of the local optimum. However, the other two algorithms are at a standstill in the later period. Owing to the impact of high-concentration pheromone, the ability of ant colony algorithm to find solutions is greatly weakened, and the algorithm cannot jump out of the local optimum. It is observed that the recommendation backtracking mechanism based on collaborative filtering can effectively help the algorithm jump out of the local optimum. What’s more, because the range of backtracking is dynamic and small, the number of iterations required to reach convergence again after backtracking is relatively small.

From the result $P$ of the rank-sum tests, we can see that most $P$ is smaller than 0.05, especially in large-scale TSP. It means that there is a significant difference in the ability to solve TSP between the improved algorithm and the traditional algorithms.

In summary, DBACS can significantly improve the convergence speed and accuracy of the algorithm in large-scale TSP.
In this paper, the standard deviation (Formula 10) is used to reflect the stability of the algorithm. The standard deviation calculated from all TSP data (15 times for each test) in the experiment is plotted as a histogram. As shown in Fig. 13, it can be concluded from the figure that the DBACS in this paper has better stability than ACS and MMAS.

\[
dev = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (l_i - l_{avg})^2} \quad (10)
\]
TABLE 10. Comparison of DBACS with latest improved algorithms.

| TSP    | DBACS | PSO-ACO-3Opt (2015) | HDACO (2018) | PACO-3Opt (2018) | DGSATSP (2014) |
|--------|-------|---------------------|--------------|------------------|----------------|
| Eu51   | 426   | 426                 | 426          | 426              | 426            |
| Berlin52| 7542  | 7542                | 7542         | 7542             | 7542           |
| St70   | 675   | 676                 | 675          | 676              | 675            |
| Eu76   | 538   | 538                 | 538          | 538              | 538            |
| Rat99  | 1211  | 1224                | 1211         | 1213             | 1211           |
| kroA100| 21282 | 21301               | 21282        | 21282            | 21282          |
| Lin105 | 14379 | 14379               | 14379        | 14379            | 14379          |

TABLE 11. Comparison of DBACS with latest improved algorithms.

| TSP    | DBACS | 4V3LGA (2017) | 1OGA (2017) | DGSATSP (2014) |
|--------|-------|---------------|-------------|---------------|
| Pr107  | 44303 | 44303         | 44574       | 44303         |
| Pr144  | 58537 | 58537         | 58537       | 58537         |
| Pr152  | 73682 | 73684         | 73684       | 73682         |
| Pr264  | 49135 | 49151         | 49236       | 49135         |
| Pr299  | 48460 | 49463         | 50093       | 48191         |

where $M$ is the test number of the algorithm in a data set; the test number in this paper is 15; $l_i$ is the solution obtained by each algorithm in the test; $l_{avg}$ is the average solution of $M$ tests in a data set.

In order to further clearly show the improvement of the improved algorithm in the accuracy of each solution, four data sets of different sizes are selected for comparative analysis. The data sets are pr107, pr226, pr439 and p654. Each data set is run 15 times with three algorithms respectively. The results are sorted from small to large, and a line chart is drawn in Fig. 14. It can be concluded from the figure that DBACS can obtain better solutions than traditional algorithms when dealing with TSP of different scales, and the stability of the algorithm is excellent. Especially in solving of large-scale pr439 and p654, the 15 solutions obtained by DBACS have higher accuracy.

The optimal paths obtained by DBACS in the test data sets have been summarized in Fig. 15.

D. COMPARATIVE ANALYSIS OF IMPROVED ANT COLONY ALGORITHM AND LATEST ALGORITHM

In order to verify the performance, the improved ant colony algorithm DBACS in this paper is compared with other latest improved algorithms, as shown in Table. 10, Table. 11 and Table. 12. The data are from the following papers. PSO-ACO-3Opt (2015) is a hybrid algorithm of the ant colony and particle swarm [20]. HDACO (2018) [32], PACO-3Opt (2018) [28], OMACO (2014) [33], and HMMA (2015) [34] are the improved ant algorithms; DGSATSP (2014) is a discrete gravitational search algorithm [35]; 4V3LGA (2017), and 1OGA (2017) are improved genetic algorithms [36]. It can be found that compared with other algorithms, DBACS has a better performance in solving TSP.

E. OVERALL PERFORMANCE ANALYSIS OF THE ALGORITHM

In this paper, ants are divided into clustering-ants and connecting-ants. In each iteration, clustering-ants are responsible for clustering, and then connecting-ants connect each small class according to the pheromone distribution to form the final path.

In the early stage of the algorithm, the DBSCAN algorithm is used to divide the current data set into several small classes according to the density. In each class, ant colony algorithm is used to construct a relatively correct path and form a favorable initial distribution of pheromones, which can accelerate the convergence speed of the algorithm. The traditional ant colony algorithms blindly search for the solution in the early stage because the difference of pheromone distribution is not obvious. Because of this, when solving large-scale TSP, the convergence speed of the algorithm is insufficient.

In the middle stage of the algorithm, due to the completion of the initial path pheromone construction, the dynamic clustering strategy plays a role in accelerating the convergence. Relying on the relatively correct pheromone distribution in the early stage, the ant colony is induced to move towards the relatively correct paths. Because the range of clustering is dynamically expanded on the original basis, the newly generated classes can obtain a good solution only by slightly
modifying some paths. The convergence speed and accuracy of the algorithm are greatly improved in this way.

At the later stage of the algorithm, due to the great difference of pheromone concentration in each path, the ant colony is seriously disturbed in the selection of paths. However, the recommendation backtracking mechanism based on collaborative filtering proposed in this paper can automatically find intensive places in the current data and carry out targeted small-scale backtracking of pheromone to get rid of the misleading of high-concentration pheromone and help the algorithm jump out of the local optimum.

To sum up, the algorithm in this paper has achieved a better balance in diversity and convergence speed, and can find better solutions in a short time.

V. CONCLUSION

This paper proposes a dynamic density clustering ant colony algorithm with a filtering recommendation backtracking mechanism. The improved algorithm includes two strategies. The first strategy is the dynamic density clustering strategy, which is used to accelerate the convergence of the algorithm and improve the accuracy of the solution. The second strategy is a recommendation backtracking mechanism based on collaborative filtering, which is used to increase population diversity and help the algorithm jump out of the local optimum. This strategy can weaken pheromone dynamically in dense data sets to reduce the negative impact of positive feedback of ant colony algorithm in the later period, so as to help the algorithm jump out of the local optimum. The experimental results show that, in large-scale TSP instances, the algorithm can accelerate the convergence speed and increase the diversity of the population, thus improving the quality of the solution.

Future research directions:
1. We will continue to study the clustering ant colony algorithm to solve larger TSP instances.
2. Combine Ant Colony Optimization with practical applications, such as robot path planning.

REFERENCES

[1] C. Mu, J. Zhang, Y. Liu, R. Qu, and T. Huang, “Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks,” Soft Comput., vol. 23, no. 23, pp. 12683–12709, Dec. 2019.
[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.
[3] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 256–279, Jun. 2004.
[4] Z. Xu, Y. Wang, S. Li, Y. Liu, Y. Todo, and S. Gao, “Immune algorithm combined with estimation of distribution for traveling salesman problem,” IEEE Trans. Electron. Electron. Eng., vol. 11, pp. S142–S154, Jun. 2016.
[5] S. Gao, W. Wang, H. Dai, F. Li, and Z. Tang, “Improved clonal selection algorithm combined with ant colony optimization,” IEEE Trans. Hum. Syst., vol. 91, no. 6, pp. 1813–1823, Jun. 2008.
[6] H. Dai, Y. Yang, C. Li, J. Shi, S. Gao, and Z. Tang, “Quantum interference crossover-based clonal selection algorithm and its application to traveling salesman problem,” IEICE Trans. Syst. Inf., vol. 92, no. 1, pp. 78–85, Jan. 2009.
[7] M. Dorigo, V. Maniezzo, and A. Colomi, “Ant system: Optimization by a colony of cooperating agents,” IEEE Trans. Syst. Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.
[8] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006.
[9] Q. Luo, H. Wang, Y. Zheng, and J. He, “Research on path planning of mobile robot based on improved ant colony algorithm,” Neural Comput. Appl., vol. 32, no. 6, pp. 1555–1566, Mar. 2020.
[10] S. Takitou and A. Taneda, “Ant colony optimization for predicting RNA folding pathways,” Comput. Biol. Chem., vol. 83, Dec. 2019, Art. no. 107118.
[11] Z. Yun, S. Shao, and T. Mai, “Solving TSP problem in precision agriculture soil survey,” China Rural Water Hydrophys, to be published.
[12] R. S. Parpinelli, H. L. Poles, and A. A. Freitas, “Data mining with an ant colony optimization algorithm,” IEEE Trans. Evol. Comput., vol. 6, no. 4, pp. 321–332, Aug. 2002.
[13] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 53–66, Apr. 1997.
[14] T. Stützle and H. H. Hoos, “MAX-MIN ant system,” Future Generat. Comput. Syst., vol. 16, no. 8, pp. 889–914, Jun. 2000.
[15] T. Saenpont, S. Phimolaires, and C. Lursinsap, “Combining new fast opposite gradient search with ant colony optimization for solving traveling salesman problem,” Eng. Appl. Artif. Intell., vol. 35, pp. 324–334, Oct. 2014.
[16] L. Zhang, C. Xiao, and T. Fei, “Improved ant colony algorithm based on bacterial foraging,” Comput. Eng. Sci., vol. 40, no. 10, pp. 1882–1899, 2018.
[17] M. Starzee, G. Starzee, A. Byrski, and W. Turek, “Distributed ant colony optimization based on actor model,” Parallel Comput., vol. 90, Dec. 2019, Art. no. 102573.
[18] D. Gaifang, F. Xueliang, L. Honghui, and X. Pengfei, “Cooperative ant colony-genetic algorithm based on spark,” Comput. Electr. Eng., vol. 60, pp. 66–75, May 2017.
[19] X. Meng, Z. Pian, and Z. Sheng, “Ant colony algorithm based on directional pheromone coordination,” Control Decis., vol. 28, no. 5, pp. 782–786, 2013.
[20] M. Math, K. K. Baykan, and H. Kodaz, “A new hybrid method based on particle swarm optimization, ant colony optimization and 5-opt algorithms for traveling salesman problem,” Appl. Soft Comput., vol. 30, pp. 484–490, May 2015.
[21] C.-Y. Pang, B.-Q. Hu, J. Zhang, W. Hu, and Z.-C. Shan, “Applying data clustering feature to speed up ant colony optimization,” Abstract Appl. Anal., vol. 2014, pp. 1–8, May 2014.

TABLE 12. Comparison of DBACS with latest improved algorithms.

| TSP  | DBACS  | Er  | OMACO (2014) | Er  | PACO-3Opt (2018) | Er  | PSO-ACO-3Opt (2015) | Er  | IMMA (2015) | Er  |
|------|--------|-----|-------------|-----|-----------------|-----|-------------------|-----|-------------|-----|
| FL417| 11949  | 0.74| 11991       | 1.1 | 11972           | 0.93| 11947             | 0.73| 12543       | 5.7 |
| PR439| 108036 | 0.76| 108651      | 1.3 | 108482          | 1.2 | 108530            | 1.2 | 114095      | 6.4 |
| P654 | 34936  | 0.84| 37119       | 7.1 | 35045           | 1.2 | 35052             | 1.2 | 37044       | 6.9 |
| U724 | 42983  | 2.56| 44408       | 6.0 | 42764           | 2.1 | 43172             | 3.0 | 46662       | 11.3|

VOLUME 8, 2020
[22] H. Yang, “Study on traveling salesman problem based on the improved chaos ant colony algorithm,” Adv. Mater. Res., vols. 989–994, pp. 2196–2199, Jul. 2014.

[23] P. Du, Z. Tang, and Y. Su, “An object-oriented multi-role ant colony algorithm and its TSP solution,” Control Decis., vol. 29, no. 10, pp. 1729–1736, 2014.

[24] Y. Pang, C. Zhong, and K. Cheng, “Ant colony optimization based on clustering ensemble and TSP algorithm in restricted solution space,” J. Univ. Sci. Technol. China, vol. 46, no. 9, pp. 780–787, 2016.

[25] J. Li, X. Meng, and X. Dai, “Collision-free scheduling of multi-bridge machining systems: A colored traveling salesmen problem-based approach,” IEEE/CAA J. Automatica Sinica, vol. 5, no. 1, pp. 139–147, Jan. 2018.

[26] MP-TESTDATA—The TSPLIB Symmetric Traveling Salesman Problem. [Online]. Available: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/

[27] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 2, pp. 601–614, Feb. 2019.

[28] Ş. Gülçü, M. Mahi, O. K. Baykan, and H. Kodaz, “A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem,” Soft Comput., vol. 22, no. 5, pp. 1669–1685, Mar. 2018.

[29] K. Yang, X. You, and S. Liu, “Self-adaptive double-population ant colony algorithm with entropy,” Comput. Eng. Appl., vol. 35, no. 82, pp. 114–117, 2015.

[30] W. Yong, “Hybrid Max–Min ant system with four vertices and three lines inequality for traveling salesman problem,” Soft Comput., vol. 19, no. 3, pp. 585–596, Mar. 2015.

[31] M. Zheng and M. Zhuo, “Solution for traveling salesman problem with four vertices and three lines genetic algorithm,” Comput. Eng. Appl., vol. 53, no. 14, pp. 148–154, 2017.

* * *

JIN YU was born in 1996. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include intelligent algorithms, path planning of mobile robots, and embedded systems.

XIAOMING YOU received the M.S. degree in computer science from Wuhan University, in 1993, and the Ph.D. degree in computer science from the East China University of Science and Technology, in 2007. She is currently a Professor with the Shanghai University of Engineering Science. Her research interests include swarm intelligent systems, distributed parallel processing, and evolutionary computing.

SHENG LIU received the M.S. degree in computer science from the Huazhong University of Science and Technology, in 1999, and the Ph.D. degree in computer science from the East China University of Science and Technology, in 2008. He is currently a Professor with the Shanghai University of Engineering Science. His research interests include quantum-inspired evolutionary computation, distributed parallel processing, and evolutionary computing.