



Multiple Ant Colony Algorithm Combining Community Relationship Network

Jiabo Zhao¹ · Xiaoming You¹ · Qianqian Duan¹ · Sheng Liu²

Received: 27 July 2021 / Accepted: 9 January 2022 / Published online: 18 February 2022
© King Fahd University of Petroleum & Minerals 2022

Abstract

Ant colony algorithm can better deal with combinatorial optimization problems, but it is still difficult to balance the solution accuracy and convergence speed facing large-scale TSP. Nowadays, most scholars focus on the route information of better ants for improvement, while ignoring the route information of general ants with a large base. So, this study proposes the multiple ant colony algorithm combining community relationship network (CACO) by collecting route information of all ants and constructing a route relationship network to improve the accuracy of the solution. The network is divided into a number of small communities that reflect the affinity of multiple colony ants to different cities through community detection with modularity. Within the communities, CACO use the excellent route exploration ability of the ant colony algorithm to identify high-quality route segments, integrating the pheromones of high-quality segments in the communities to provide pheromone feedback to the multiple colony ants for better route exploration. The three parts of route information collection, community detection and pheromone feedback form a feedback loop, which keeps cycling when multiple populations ants explore, and each cycle will drive the result closer to the optimal solution. Meanwhile, CACO proposes a mutual assistance strategy to improve the exploration ability of multiple colony ants by complementing each other according to the different states of superior and inferior populations. To test the performance of CACO, 28 TSP instances are compared with the well-known improved algorithms are compared and results show CACO outperforms other improved algorithms significantly, especially in large-scale TSP.

Keywords Ant colony algorithm · Modularity · Community detection · Route relation network · TSP

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

✉ Xiaoming You
yxm6301@163.com

Jiabo Zhao
1451832452@qq.com

Qianqian Duan
dq1019@163.com

Sheng Liu
1s6601@163.com

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

1 Introduction

Traveling Salesman Problem (TSP) [1] is one of the most fascinating classes of combinatorial optimization problems. It is difficult to solve by a single mathematical formulation and is often used as an acknowledged criterion to test the performance of algorithms. The TSP problem can be described as a shortest route problem in which a traveler does not repeatedly pass through all the specified cities and finally returns to the initial city. Although for small-scale cities, some mathematical methods can be used to find the route such as cutting planes algorithm [2], branch and cut scheme [3]. However, as the number of cities increases, the amount of computation when using mathematical methods increases exponentially. Especially for large-scale cities, such an approach spent huge amounts of time but did not work well, so the meta-heuristic algorithm was born. Meta-heuristic algorithms, also known as intelligent optimization algorithms, differ from optimization algorithms in that they



can construct a feature model based on a specific problem and find a feasible solution under specified constraints. Many scholars have studied meta-heuristic algorithms in order to solve different complex problems, such as genetic algorithm [4], particle swarm optimization [5], grey wolf optimization algorithm [6], whale optimization algorithm [7]. As a typical swarm intelligence algorithm, the ant colony algorithm has excellent ability to find the optimal solution in solving NP-hard problems, which has profound research value and significance.

The ant colony algorithm originated from the doctoral dissertation of M. Dorigo. He simulated the behavior of ants foraging, constructed a route and pheromone update model, and proposed the ant colony optimization algorithm (ACO) [12, 13], which achieved better results in the TSP problem. Later scholars according to their research fields applied ACO to the scheduling problem [8, 9], location routing problem [10, 11], community detection [12, 13], etc., which verify the great performance of ACO.

However, because of the positive feedback of pheromones in the ACO, pheromones will accumulate rapidly on individual routes, resulting in local optima. To alleviate the problem, M. Dorigo further proposed the ant colony system (ACS) [14]. Global and local pheromone update methods were proposed in ACS as well as a local search algorithm was added to improve the performance. Then T. Stutzle proposed the max–min ant system (MMAS) [15], which avoids excessive pheromone accumulation by setting a maximum–minimum threshold of pheromone to control the pheromone concentration within a certain range.

Although these above methods have good performance, they still face problems such as easy to fall into local optimum and slow convergence speed. For these problems, many scholars have proposed their improvement strategies.

The performance improvement of ant colony algorithm is mainly based on the convergence speed and solution accuracy. Zhang, Q. et al. [16] accelerated convergence by reinforcing dynamic pheromone edges. Zhao, D. et al. [17] enhanced convergence speed by cross-searching horizontally and allowing two kinds of ants to exchange information. Li, S. et al. [18] defined a collective action strategy where each ant decides whether to participate in the collective action based on threshold to increase the diversity of the algorithm. Wu, Y. et al. [19] introduced a differential evolution operator to increase the global search capability by modifying the update of the mutation operator.

Besides, some scholars have modified the characteristic model of ACO algorithm. Tam, J.H. et al. [20, 21] optimized the parameters of ACO algorithm by PSO algorithm to improve the solution quality. Abdelbar, A.M. et al. [22, 23] changed the fixed parameters into adjustable parameters to better adapt to different stages of the ant colony algorithm for increasing the accuracy of the solution. Wang, M. et al. [24,

25] added a specified function to the pheromone to improve the solution of the optimal problem. Deng, W. et al. [26] proposed a collaborative strategy based on the crossover variation of genetic algorithm to improve the poor local search ability. Dai, X. et al. [27] combined ACO algorithm with the evaluation function of A* algorithm to enhance the heuristic role of ACO to speed up the convergence. Mohsen, A.M. et al. [28] incorporated the variation factor of the simulated annealing algorithm to increase the diversity of ACO algorithm for more efficient route exploration.

With the increase in city nodes, some scholars found that multiple ant colony [29–32] have better performance than single colony. Gong, Y. et al. [30] performed a single-objective search with improved pheromone update rules for two colonies and used a heuristic strategy to balance multi-objective optimization to improve algorithm performance. Xu, M. et al. [32] let two populations be, respectively, responsible for solution diversity and convergence speed, and exchange information during exploration to ensure the balance between diversity and convergence speed. Although these improved multiple ant colony algorithms above have good optimization effect, they mostly exchange high-quality information between populations to optimize the algorithms, and the route information of general ants is less studied. But is the route information of a huge number of general ants really worthless? To answer the question, we conducted the present study.

We find that the reason why ants cannot find a better solution in solving TSP is that they fall into local optimum, especially for large-scale cities. When ants explore the solution, the routes taken by elite ants are dominated, making a large accumulation of pheromones and thus failing to find a better solution. To address this situation, we use the routes of general ants as an important reference as well to try to alleviate. We refer to the community detection algorithm [33, 34] to construct the route relationship network and collect the route information of all ants in different populations, finding that the route information of ants in general is extremely diverse, which can improve the situation of getting stuck in stagnation. Meanwhile, the nodes are also divided by the criterion of modularity [35], and the obtained communities balance diversity and stability.

Although some scholars have previously combined community relations with ant colony algorithms, they are used for community detection rather than solving the TSP problem. Noveiri, E. et al. [36] combined pheromones with modularity to find an outer region and then implemented community detection by fuzzy clustering. Romdhane, L. et al. [37] defined an objective function of set purity and density based on the community structure and used the ant colony algorithm to optimize the objective function to classify quality communities that can balance the two metrics. Although they also combined the ant colony algorithm, they did not involve

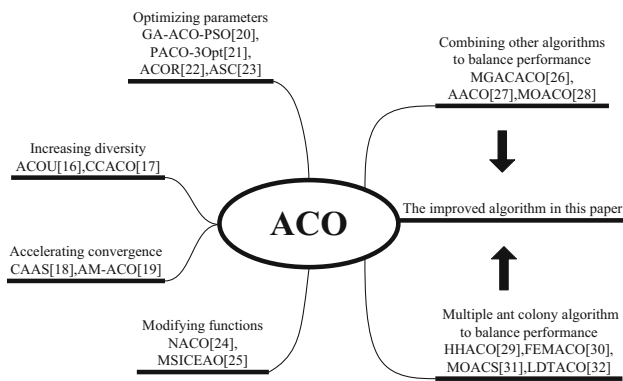


Fig. 1 The improved algorithms based on ACO

the study of node sparsity and node distance. In this paper, we will improve the ant colony algorithm by incorporating the idea of community detection to balance the relationship between diversity and convergence speed, and propose the multiple ant colony algorithm combining community relationship network to solve the TSP problem. The improved algorithms based on ACO is shown in Fig. 1.

The main contributions and innovations of this study are as follows.

Firstly, we integrate the idea of community detection and sample the route information of all ants. The route relationship network containing node sparsity and distance relationship is constructed by sample processing. Then, we divide the route relationship network into communities with the criterion of modularity, because this approach does not just consider the distance, so the divided communities will increase the diversity of the ant colony algorithm.

Secondly, route exploration is performed with ant colony algorithm in the divided small communities. Benefiting from the structural stability of community detection, routes in small communities have the potential to explore optimal solutions, which can enhance the convergence speed of the algorithm when pheromone guidance is given to multiple swarms. Also, pheromone guidance of non-optimal routes for multiple colonies of ants will increase the diversity and further improve the algorithm performance.

Finally, because of the pheromone guidance of small communities, the exchange of information among multiple colonies during exploration is not limited to distance information, and the sparse relationship of general ants for nodes is also considered. Therefore, the superiority and inferiority mutual aid strategy is used when multiple colonies exchange information to further improve the accuracy of the solution.

The remainder of this study is as follows: Sect. 2 describes ACS, MMAS and community detection algorithms. Section 3 describes in detail the improvement algorithm. Section 4 presents the experimental results of the improved algorithm.

Finally, Sect. 5 concludes the work of this study and provides an outlook on future research directions.

2 Related Work

2.1 Ant Colony System

The formula for route construction of ACS is as follows.

$$S = \begin{cases} \arg \max_{j \in \text{allowed}} \{ \tau_{ij} \cdot \eta_{ij}^\beta \}, & q \leq q_0 \\ s, & \text{else} \end{cases} \quad (1)$$

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in \text{allowed}} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta}, & j \in \text{allowed} \\ 0, & \text{else} \end{cases} \quad (2)$$

where $\tau_{ij}(t)$ represents the magnitude of the pheromone between cities i and j at moment t . The starting concentration on each edge is the same, denoted as τ_0 . η_{ij} represents the reciprocal of the distance d_{ij} between cities i and j , and $\eta_{ij} = 1/d_{ij}$. q_0 is a constant value in the interval $[0,1]$, q is a random number in the interval $[0,1]$, and $j \in \text{allowed}$ indicates that city j belongs to the set of optional cities outside the taboo table. S denotes the next city that will be selected when $q \leq q_0$ and s is the selection method of the roulette wheel of Eq. (2).

The update mechanism of ACS is divided into two parts: global pheromone update and local pheromone update.

- *Global pheromone update:* After all ants have completed research, the algorithm only updates the pheromones on the current optimal route. Its calculation formula is as follows.

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (3)$$

$$\Delta\tau_{ij} = \begin{cases} 1/L_{gb}, & (i, j) \in \text{Best Tour} \\ 0, & \text{else} \end{cases} \quad (4)$$

where ρ is the global pheromone evaporation rate. $\Delta\tau_{ij}$ is the pheromone increment. L_{gb} is the length of current optimal route.

- *Local pheromone update:* Once all ants have completed their iterations, the algorithm performs a local pheromone update for all routes. This shortens the pheromone difference between optimal and non-optimal routes and increases the diversity of the algorithm. The formula is as follows.

$$\tau_{ij}(t + 1) = (1 - \xi)\tau_{ij}(t) + \xi\tau_0 \quad (5)$$

where τ_0 is the starting concentration on each edge and ξ is the local pheromone evaporation rate.

2.2 Max–Min Ant System

The main idea of MMAS is to control the pheromone range in a certain interval such that $\tau \in [\tau_{\min}, \tau_{\max}]$. When $\tau_{ij} < \tau_{\min}$, make $\tau_{ij} = \tau_{\min}$; when $\tau_{ij} > \tau_{\max}$, make $\tau_{ij} = \tau_{\max}$. Where τ_{\min}, τ_{\max} are calculated according to Eqs. (6) and (7).

$$\tau_{\max} = (1/\rho) \cdot (1/T^{\text{gb}}) \tag{6}$$

$$\tau_{\min} = \tau_{\max}/2n \tag{7}$$

where T^{gb} is the length of global optimal route and ρ is the pheromone evaporation rate.

Based on the limitation of the pheromone range, only the pheromones of the current or global optimal route are updated at the end of each iteration according to Eqs. (8) and (9). This allows the algorithm to explore the routes more efficiently and increases the diversity of the algorithm.

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}} \tag{8}$$

$$\Delta\tau_{ij}^{\text{best}} = 1/f(s^{\text{best}}) \tag{9}$$

where $f(s^{\text{best}})$ is the current optimal route or the global optimal route.

2.3 Community Detection

2.3.1 Representation of Complex Networks

Complex networks are generally represented as graphs with nodes and edges, such as $G = (V, E)$, where $V = \{1, 2, \dots, n\}$ denotes the set of nodes and $E \subseteq V \times V$ denotes the set of edges. The nodes and edges are usually represented with the help of the adjacency matrix $A_{n \times n}$ when they are processed. Figure 2 shows the community relationship network of 34 members of the karate club community. Equation (10) shows the weights of nodes and edges

$$A_{n \times n} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{10}$$

2.3.2 Modularity

The modularity was proposed by Newman to address the notion that the Girvan and Newman algorithm (GN) cannot evaluate the quality of a community. It was widely used to

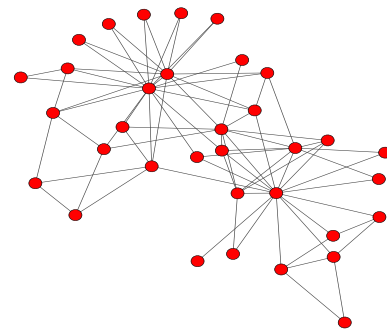


Fig. 2 Karate club relationship network

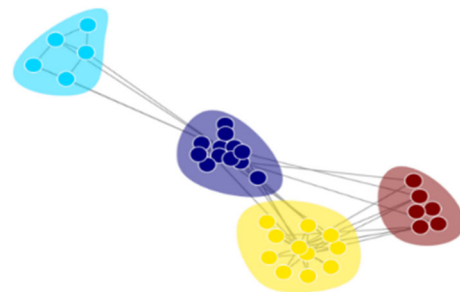


Fig. 3 Relationship network after community detection

measure the strength of the community structure after network segmentation, as shown in Eq. (11).

$$Q = \sum_i^K (e_{ii} - a_i^2) = \frac{1}{2m} \sum_v^n \sum_w^n \left(A_{vw} - \frac{d_v d_w}{2m} \right) I(c_v, c_w) \tag{11}$$

where K is the number of communities. e_{ii} is the proportion of edges inside community i to the total number of edges. $a_i = \sum_{j=1}^K e_{ij}$ is the proportion of edges connected to community i to the total number of edges, then a_i^2 denotes the proportion of edges in community i to the total number of edges. m is the total number of edges in the network. n is the total number of nodes in the network. A_{vw} is the value of the adjacency matrix of node v and w . $d_v = \sum_{w=1}^n A_{vw}$ is the degree of node v . $I(c_v, c_w)$ is the schematic function, $I(c_v, c_w) = 1$ when node v and node w belong to the same community, otherwise $I(c_v, c_w) = 0$.

By dividing the karate club relationship network in Fig. 2 through modularity, we can see that the 34 members are divided into four types of communities, which are closely related within the community and loosely related between the communities in Fig. 3.

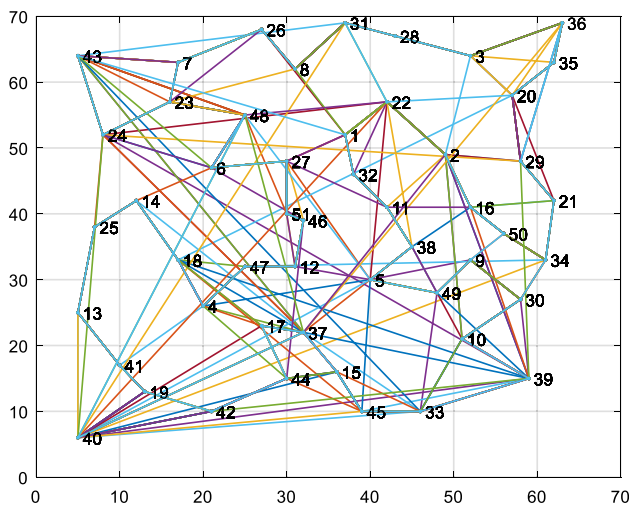


Fig. 4 Route relation network of TSP city set Eil51

3 Proposed Algorithm

3.1 ACO Combined with Community Detection

When the ant colony algorithm uses multiple populations to solve the TSP problem, the communication and collaboration among multiple populations is a great advantage to improve the performance of the algorithm, but most scholars have achieved the improvement of the algorithm performance by exchanging the pheromones and optimal routes among different populations, and there is little research on the social relationship network when all ants of different populations explore the routes. Therefore, this study proposes a community collaboration strategy to study the route structure relationship network among populations with the help of community detection methods from the route relationship network when different populations explore routes, and apply it to solve the TSP problem. In this section, we describe how to construct the route relationship network for the ant colony algorithm.

3.1.1 Route Relationship Network Model

The route network in this study, is a synthetic network that collects the route information provided by all ants of various colonies. As shown in Fig. 4, the route relationship network constructed when ants explore the TSP city set Eil51, each node represents a city and the connection between nodes is an ant route.

3.1.2 Processing of Route Information Samples

As shown in Fig. 4, the route relationship network has a certain complex network structure. With 51 cities as nodes and the exploration routes of all ants as edges, it can be clearly

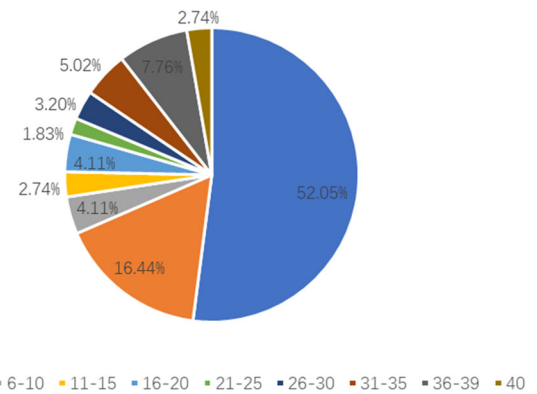


Fig. 5 Graph of the weight share of connected edges

found that some nodes have a certain correlation structure with a significantly dense number of connected edges, while some nodes have few connected edges. In Fig. 4, the city with the number 40 in the lower left corner has more connections with other cities, which proves that this city has very diverse route choices and is a more difficult point to classify in community detection. In contrast, the city with the number 13 has fewer connections to other nodes. Most ants will follow these routes, which are more likely to be high quality routes and easier to classify in community detection, with relatively stable community structure.

Classical datasets such as the Karate dataset only have information about nodes and edges, and connecting edges is not considered for distance, so special treatment is needed to solve the TSP problem. In the ant colony algorithm, ants taking the same route is a sure way to explore better routes. For this problem, this study modifies the adjacency matrix reflecting the route information to be more consistent with the TSP problem by weighting the repeated routes when collecting the route information. When the route sample information is collected and transformed into the adjacency matrix, the adjacency matrix is changed into a weighting matrix by Eq. (12). When node i is not connected to node j , make $a_{ij} = 0$; when there is a connection, then make a_{ij} increment.

$$a_{ij} = \begin{cases} 0, & \text{if } i \text{ and } j \text{ are not connected} \\ a_{ij} + 1, & \text{if } i \text{ and } j \text{ are connected} \end{cases} \quad (12)$$

As shown in Fig. 5, this study used 40 ants for 20 independent sample collections on the city set Eil51, and found that the proportion of connected edges with a weight of 1 was more than 50%, and a total of nearly 70% on the proportion of weights with low weights 1 to 5, which shows that the route network has a more complex relational structure. Also, due to the positive feedback of pheromones, it is not the case that the proportion decreases with higher weights, but starts to increase with higher weights. This reflects that some ants are following the quality ants when exploring.

In order to avoid affecting the quality of community division by large differences between the upper and lower limits of weights, we use Eqs. (13) and (14) to set an upper weighting limit for the connected edges, which makes the internal differences of the adjacency matrix narrow and increases the diversity of the algorithm.

$$(1 - v) * \text{SUM} \rightarrow A_{\max} \quad (13)$$

$$A_{ij} = \begin{cases} A_{ij}, & \text{if } A_{ij} < [A_{\max}] \\ [A_{\max}], & \text{if } A_{ij} \geq [A_{\max}] \end{cases} \quad (14)$$

where SUM is the number of connected edges in the adjacency matrix. v is the filtering rate to set the upper limit of the weights. From Fig. 5 we can see that the value of 10% is more appropriate. $[A_{\max}]$ meets the number of weights corresponding to the filtering rate, as the upper limit of the weights, and $[x]$ denotes the largest integer no more than x .

3.2 Community Collaboration Strategy

Classical community detection algorithms deal with the relationship between nodes and edges, while the TSP problem is an optimization problem involving distance. This section details how to take advantage of the community structure for better guiding ants to find routes.

Combining Figs. 4 and 5, it can be found that dividing the community is not done just by considering the distance. For city nodes with multiple connected edges, it has a rich diversity of choosing other cities. While for city nodes with fewer connected edges, it indicates that this is the high-quality route that most ants will choose, which can speed up the convergence speed. Therefore, the divided communities satisfy the requirements of the ant colony algorithm for diversity and convergence speed.

As shown in Figs. 6 and 7, firstly, different ant colonies explore the TSP city set according to their own characteristics. When the route information is mature, the route information of each population is collected as samples to construct a route relationship network model. Then the route relationship network is divided into several structurally stable small communities according to the modularity, and in the small communities we use ACS for route exploration to find the shortest closed-loop routes and pheromones.

Then, we use high-quality pheromones from small communities to guide the inferior populations in multiple populations for improving the search quality of inferior ants. Meanwhile, the superior and inferior populations exchange information to get out of their respective dilemmas when they fall into local optimum.

The three components of collecting route information, community detection, and pheromone guidance are cycled several times to further explore better solutions. In this study,

samples were collected once every 500 iterations. In Fig. 6, $A_{n \times n}$ is the adjacency matrix of the route relationship network model, and $P_{n \times n}$ is the pheromone matrix of each community after merging.

3.2.1 Advantageous Route Guidance

When processing the route samples, an upper weight limit is set for the edges to ensure the quality of community detection. However, the route with the highest weight is the route chosen by most ants and is likely to find the best solution.

Therefore, we enhanced the guidance of high-quality routes when collecting pheromones of optimal routes in small communities to highlight the difference between high-quality routes and common routes. After the sample collection was completed, the weights of the routes were sorted in descending order. In the global pheromone update, the pheromones of the top three edges with the highest weights are guided by Eq. (15) to improve the efficiency of the ants in exploring better solutions.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho(1/L_{\text{gb}} + W/(n \cdot d_{\text{max}})) \quad (15)$$

where ρ is the global pheromone evaporation rate. L_{gb} is the length of current optimal route. W is the edge weight. d_{max} is the maximum city distance, and n is the number of cities.

3.2.2 Pheromone Guidance

In dealing with how to solve TSP problems using small divided communities, we use a pheromone guided strategy to complete. We chose to exploit the properties of ACS within small communities to find high-quality solutions faster. Although not all routes explored within each small community are optimal solutions, there exist some routes that are high-quality pieces of optimal solutions.

In summary, this study integrates the pheromones of each small community to integrate a set of high-quality route pheromones, and provides pheromone guidance for multiple groups of ants according to Eqs. (16–18) to guide multiple groups to explore better routes, and also provides samples with better information for the next sample collection of community detection.

$$\text{Pheromone}_i = (1 - \sigma) \cdot \text{Pheromone}_i + \sigma \cdot K \cdot \text{Pheromone} \quad (16)$$

$$K = \begin{cases} 1, & \text{if inferior population is ACS} \\ r, & \text{if inferior population is MMAS} \end{cases} \quad (17)$$

$$r = \frac{(\tau_{\max}^{\text{MMAS}} - \tau_{\min}^{\text{MMAS}})}{(\tau_{\max}^{\text{ACS}} - \tau_{\min}^{\text{ACS}})} \quad (18)$$

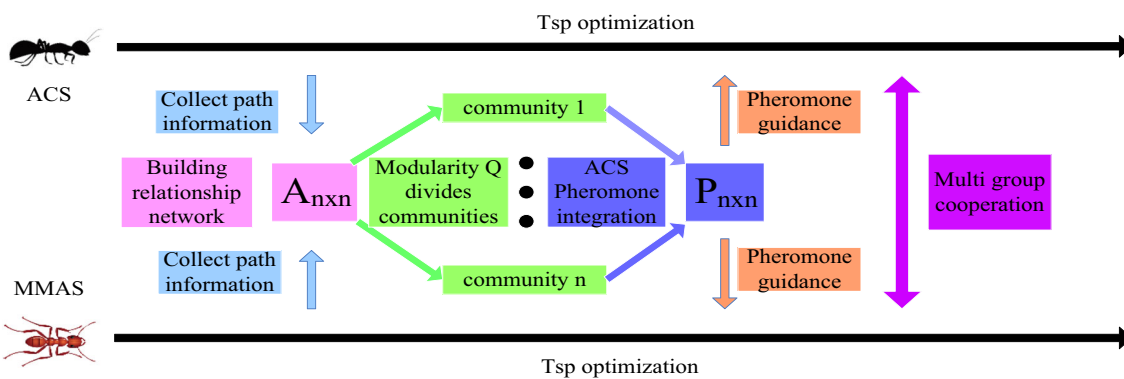


Fig. 6 Schematic diagram of community collaboration strategy

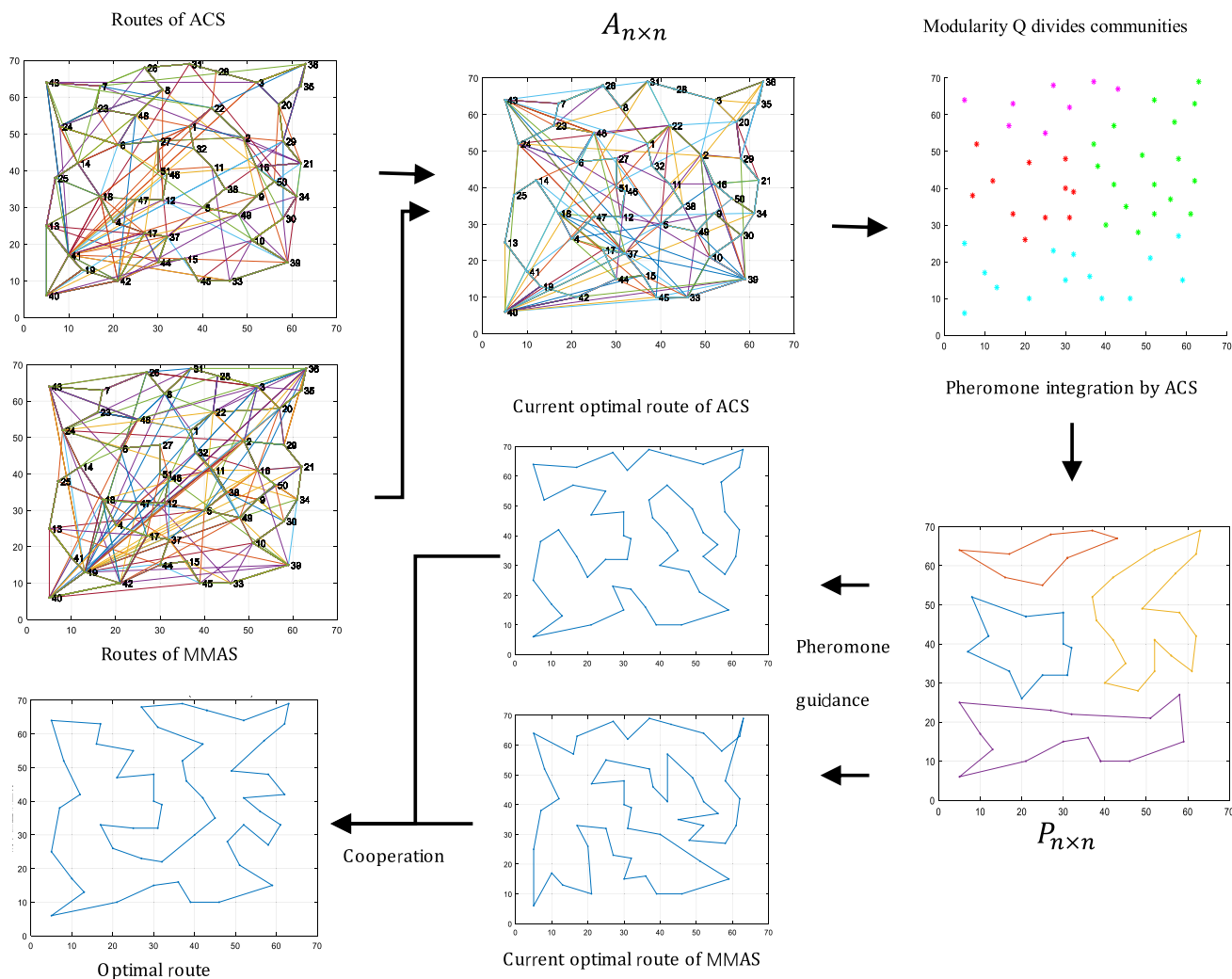


Fig. 7 Steps of community collaboration strategy

where Pheromone_i is the pheromone of inferior population. Pheromone_c is the combined pheromone collected in the small communities. σ is the guidance weight, for the inferior population, the current optimal route is less effective and should be guided by a slightly larger weight, so $\sigma = 0.6$. K is the conversion coefficient. The transformed pheromone will be more reasonable because the MMAS population has a pheromone range limit, the upper and lower limits of which are different from those of the pheromone of ACS. τ_{max}^{MMAS} and τ_{min}^{MMAS} are the upper and lower limits of the pheromone of MMAS. τ_{max}^{ACS} and τ_{min}^{ACS} are the upper and lower limits of the pheromone of ACS.

3.2.3 Superior and Inferior Mutual Assistance Strategies

When multiple populations consisting of ACS and MMAS exchanged information, the superior population explored the routes with higher precision, but got stuck in stagnation more severely. At this time, although the routes of the inferior population are not very good, their pheromones have better diversity compared with the dominant population. Feeding the pheromones of the inferior population to the superior population, which would increase the possibility of the superior population jumping out of stagnation.

Based on the above principles, we propose superior and inferior mutual assistance strategies, according to Eqs. (19) and (20) to exchange information of multiple populations. The route information of the superior population is transmitted to the inferior population, while the inferior population feeds back the corresponding pheromones to help the superior population explore a better solution.

$$L_{best}^{inferior} = \begin{cases} L_{best}^{superior}, & \text{if algorithm is stuck} \\ L_{best}^{inferior}, & \text{if algorithm is not stuck} \end{cases} \quad (19)$$

$$\tau^{superior} = \begin{cases} \omega \cdot \tau^{superior} + (1 - \omega)r \cdot \tau^{inferior}, & \text{if inferior is ACS} \\ \omega \cdot \tau^{superior} + (1 - \omega)\frac{1}{r} \cdot \tau^{inferior}, & \text{else} \end{cases} \quad (20)$$

where $L_{best}^{inferior}$ is the shortest route of the inferior population. $L_{best}^{superior}$ is the shortest route of the superior population. $\tau^{superior}$ is the pheromone of the superior population. ω is the feedback ratio, and the good route information of the dominant population is also taken into account in the feedback, so $\omega = 0.7$.

3.3 Algorithm Framework

The algorithm framework is shown in Fig. 8, and the orange part is the improvement that differs from the classical algorithm. First, ACS and MMAS jointly explore

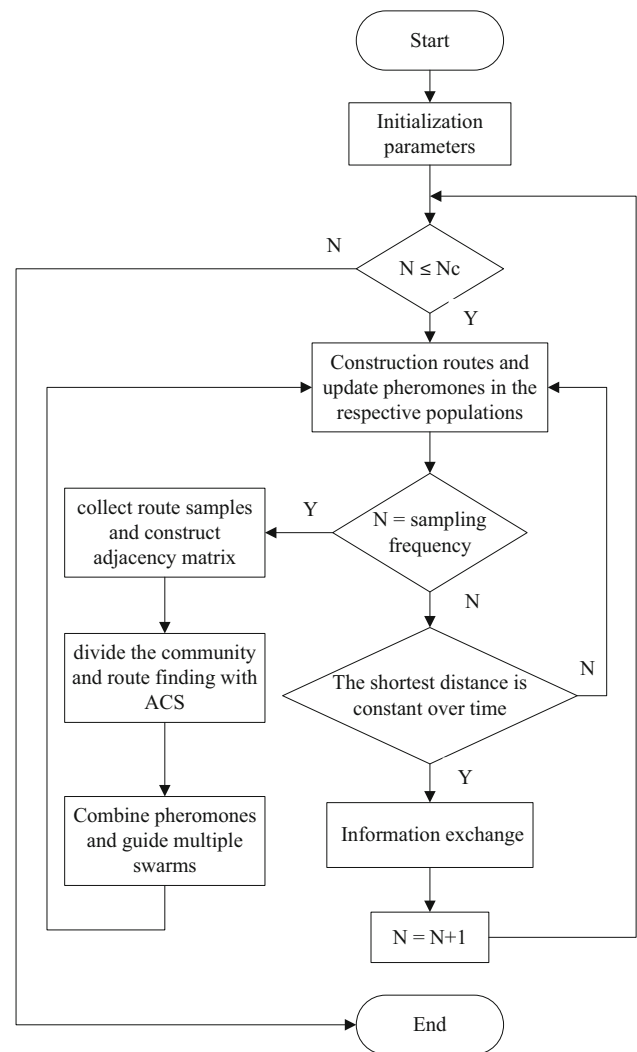


Fig. 8 Algorithm framework

the optimal routes, and when the sampling frequency is reached, we collect mature route samples to construct a route relationship network with multiple populations. Then we use community detection to delineate small communities with potential quality routes and use ACS for route search.

Afterward, the pheromones of each small population are integrated and pheromone guidance is applied to multiple populations to form a closed loop of feedback which adjusts the meritocracy of the inferior populations. If the shortest distance is constant over time, the algorithm is judged to be in stagnation, at which point the superior and inferior mutual assistance strategies is used to exchange information and increase the chance of jumping out of stagnation.

Algorithm CACO for TSP.

1. **Initialize** the pheromone and the parameters
2. calculate the distance between cities
3. calculate the initial pheromone matrix
4. **Initialize** iteration $N=0$, $N_c=2000$
5. **while** $N < N_c$ **do**
6. construct ant solutions for ACS, MMAS by equation (1) and (2)
7. update the pheromone matrix for ACS, MMAS by equation (3) to (9)
8. **if** N satisfies the sampling frequency
9. collect route samples and strengthen the pheromone of advantage route by equation (15)
10. construct adjacency matrix by equation (10) and (12) to (14)
11. divide the community by equation (11)
12. guiding multi population with community pheromone by equation (16) to (18)
13. **end if**
14. **if** the shortest distance of multiple swarms remains constant over time
15. Information exchange between superior and inferior populations by equation (19) and (20)
16. **end if**
17. $N=N+1$
18. **end while**
19. **return** the best solution

4 Experiment and Simulation

To test the performance of the CACO algorithm, this study uses a simulation environment of Windows 10 system, matlab2016a version, and selects multiple data sets of international standard TSP database for simulation experiments. The number of ants per population is 20, and the maximum number of iterations is 2000 iterations.

4.1 Parameters Setting

4.1.1 ACS and MMAS Parameter Setting

For the ACS algorithm, we conduct the orthogonal experiments with five factors and four levels on the city set Eil51. To ensure the stability of the parameters, data were collected 20 times for each set of parameters and the average solution was taken. Finally, the optimal parameters of the ACS algorithm were determined to be: $\alpha = 1$, $\beta = 4$, $\rho = 0.1$, $\xi = 0.3$, $q_0 = 0.8$.

Similarly, for the MMAS algorithm, the optimal parameters are: $\alpha = 1$, $\beta = 5$, $\rho = 0.1$.

4.1.2 Parameter Setting for Sample Size

The construction of the route relationship network depends on the collected route information samples, and the number of samples will have a large impact on the modularization performance [38]. In the ant colony algorithm, the number of samples that can be collected in one iteration is a fixed value. For larger cities, it is possible to collect a sparse network of route relationships, making most of the nodes have thin or even no relationships with each other, which results in the inability to form a complete network of relationships. Therefore, we use the minimum route length, the average route length and the average module degree as criteria to determine the parameters by increasing the number of iterations.

The experiments were conducted with 20 ants each from ACS and MMAS, and samples were collected in Pr76 city set for iteration numbers from 1 to 10. To ensure the accuracy of the experiments, 20 independent experiments were conducted. The experimental results are shown in Table 1 and



Table 1 Pr76 sample collection statistics

$iter_c$	L_{best}	L_{ave}	Q_{ave}
1	108,159	109,669.15	0.721
2	108,304	109,592.35	0.718
3	108,159	109,715.30	0.715
4	108,274	109,661.80	0.717
5	108,159	109,531.80	0.713
6	108,304	109,732.05	0.707
7	108,304	109,807.75	0.713
8	108,159	109,708.25	0.708
9	108,234	109,641.40	0.712
10	108,234	109,418.75	0.713

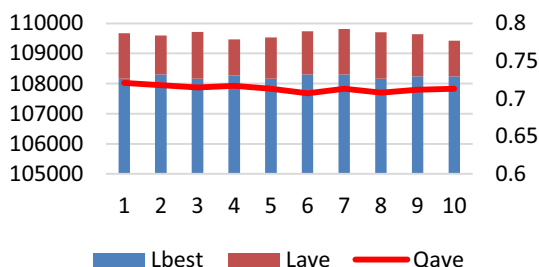
**Fig. 9** Comparison of Pr76 sample collection

Fig. 7, where $iter_c$ is the number of sample collection iterations, L_{best} is the minimum route length, L_{ave} is the average of the route length, and Q_{ave} is the average of modularity.

From Fig. 9, the modularity from one to ten iterations does not vary much and is stable at about 0.71, because the ant routes are different in each iteration instead of repeating only one route, so the structure of the route relationship network is relatively stable even for multiple iterations.

Observing the minimum route lengths in Table 1, we can find that the optimal solution for the city set Pr76 is found at iteration numbers 1,3,5,8, when the exploration performance is better. From the average length, the number of iterations is in the local minimum position at 5. Although there is also a tendency to reduce after 10 iterations, it will lead to collecting too many samples and make the route relationship network structure more complicated. The above analysis finally determines the number of sample collection iteration $iter_c$ is 5.

4.2 Comparison with Traditional Ant Colony Algorithm

To demonstrate the performance of the improved algorithm, we used the TSP, which is recognized to measure the performance of discrete optimization algorithms, as the standard for our experiments. In this section, the performance of the algorithm is analyzed by applying the ACS, MMAS and CACO algorithms to the TSP examples of 28 city sets and compar-

ing the optimal solution L_{best} , the average solution L_{ave} , the minimum error rate E_{min} and the standard deviation STD through 20 independent experiments.

The error rate is calculated by Eq. (21) and the average modularity is calculated by Eq. (22). Where L_{opt} is the standard optimal solution of the city set. L_{best} is the optimal solution of the algorithm. L_i is the final distance of 20 experiments, and L_{ave} is the average of the final distance of 20 experiments. These experimental results are shown in Table 2 and Fig. 10.

$$E_{min}(\%) = \left(\frac{L_{best} - L_{opt}}{L_{opt}} \right) \times 100\% \quad (21)$$

$$STD = \sqrt{\frac{1}{N} \sum_{i=1}^N (L_i - L_{ave})^2} \quad (22)$$

As shown in Table 2 and Fig. 10, all three algorithms find optimal solutions in the small-scale city sets Eil51, St70, and KroA100. In the medium-scale city sets with the number of cities between 100 and 200, the improved algorithm can find better or optimal solutions for the city set with higher solution accuracy compared with the traditional algorithm, in which the optimal solutions are found for the Ch130 and KroA200 city sets. In the large-scale city sets with more than 200 cities, the improved algorithm can clearly show that the accuracy of the solution is better than the traditional algorithm, and it can keep the minimum error rate within 1%, so it has a better ability of finding the optimal solution.

From (b), (d), (f) and (h) in Fig. 10, we can see the convergence speed and minimum error of CACO. It can find higher precision solutions in a shorter number of iterations and converge quickly in a shorter number of iterations with the pheromone guidance strategy of community detection in the early stage. Later, through the information exchange and mutual assistance of multiple communities, it can realize to jump out of stagnation and further improve the solution accuracy, which better balances the relationship between solution accuracy and convergence speed in the ant colony algorithm.

4.3 Comparison with the Latest Improved Algorithm

To verify the performance of the CACO algorithm, 20 independent experiments were conducted in different size city sets and compared with the latest improved particle swarm optimization algorithm MPSO [39] and wolf swarm optimization algorithm D-GWO [40] so far, and the results are shown in Tables 3 and 4.

As shown in Table 3, both CACO and MPSO can find the optimal solution in the small-scale city sets with the number of cities around 100, but the quality of the optimal solution gradually decreases with the expansion of the city size, starting from KroB150 for MPSO, and the error rate of the optimal



Table 2 Performance comparison of cities data sets of different sizes

TSP	L_{opt}	Algorithm	L_{best}	L_{ave}	$E_{min} \%$	STD
Eil51	426	ACS	426	428.15	0.00	2.10
		MMAS	426	428.05	0.00	2.09
		CACO	426	427.10	0.00	0.86
Berlin52	7542	ACS	7542	7543.50	0.00	0.51
		MMAS	7542	75,423.00	0.00	0.51
		CACO	7542	7542.00	0.00	0
St70	675	ACS	675	679.10	0.00	3.49
		MMAS	675	677.60	0.00	2.71
		CACO	675	677.15	0.00	2.56
KroA100	21,282	ACS	21,282	21,433.60	0.00	191.87
		MMAS	21,282	21,396.30	0.00	166.58
		CACO	21,282	21,337.20	0.00	45.44
KroB100	22,141	ACS	22,246	22,311.05	0.47	29.98
		MMAS	22,220	22,320.45	0.36	91.42
		CACO	22,141	22,235.00	0.00	31.69
Ch130	6110	ACS	6146	6220.45	0.59	65.11
		MMAS	6121	6188.60	0.18	37.54
		CACO	6110	6139.15	0.00	8.36
Ch150	6528	ACS	6554	6591.55	0.40	42.32
		MMAS	6548	6578.65	0.31	29.55
		CACO	6532	6548.70	0.06	6.33
KroA150	26,524	ACS	26,664	27,108.95	0.53	222.60
		MMAS	26,665	27,010.80	0.53	177.04
		CACO	26,583	26,708.85	0.22	54.90
KroA200	29,368	ACS	29,440	29,646.62	0.25	242.34
		MMAS	29,460	29,645.45	0.31	185.09
		CACO	29,368	29,501.55	0.00	45.15
KroB200	29,437	ACS	29,819	30,194	1.30	261.93
		MMAS	29,766	30,197.95	1.12	401.22
		CACO	29,524	29,834.05	0.30	137.41
Tsp225	3916	ACS	3933	3997.55	0.43	39.42
		MMAS	3939	3984.15	0.59	31.63
		CACO	3923	3978.30	0.18	15.51
Pr226	80,369	ACS	80,867	59,563.90	0.62	544.26
		MMAS	80,819	59,353.45	0.56	513.30
		CACO	80,373	80,940.90	0.01	442.81
A280	2579	ACS	2605	2642.60	1.01	36.48
		MMAS	2597	2629.75	0.70	27.75
		CACO	2583	2607.65	0.16	13.56
Lin318	42,029	ACS	43,203	43,626.20	2.79	221.68
		MMAS	43,046	43,615.45	2.42	447.74
		CACO	42,365	42,878.45	0.80	194.83

Table 2 continued

TSP	L_{opt}	Algorithm	L_{best}	L_{ave}	$E_{min} \%$	STD
Fl147	11,861	ACS	12,192	12,185.25	2.80	100.58
		MMAS	12,142	12,174.20	2.37	94.02
		CACO	11,950	12,018.20	0.75	25.90
Pr439	107,217	ACS	109,151	110,906.25	1.80	2402.69
		MMAS	108,935	110,826.45	1.60	1682.36
		CACO	107,883	108,683.20	0.62	345.02
P654	34,643	ACS	35,551	35,476.70	2.62	535.75
		MMAS	36,048	35,996.85	4.06	518.65
		CACO	34,932	35,662.90	0.83	483.66

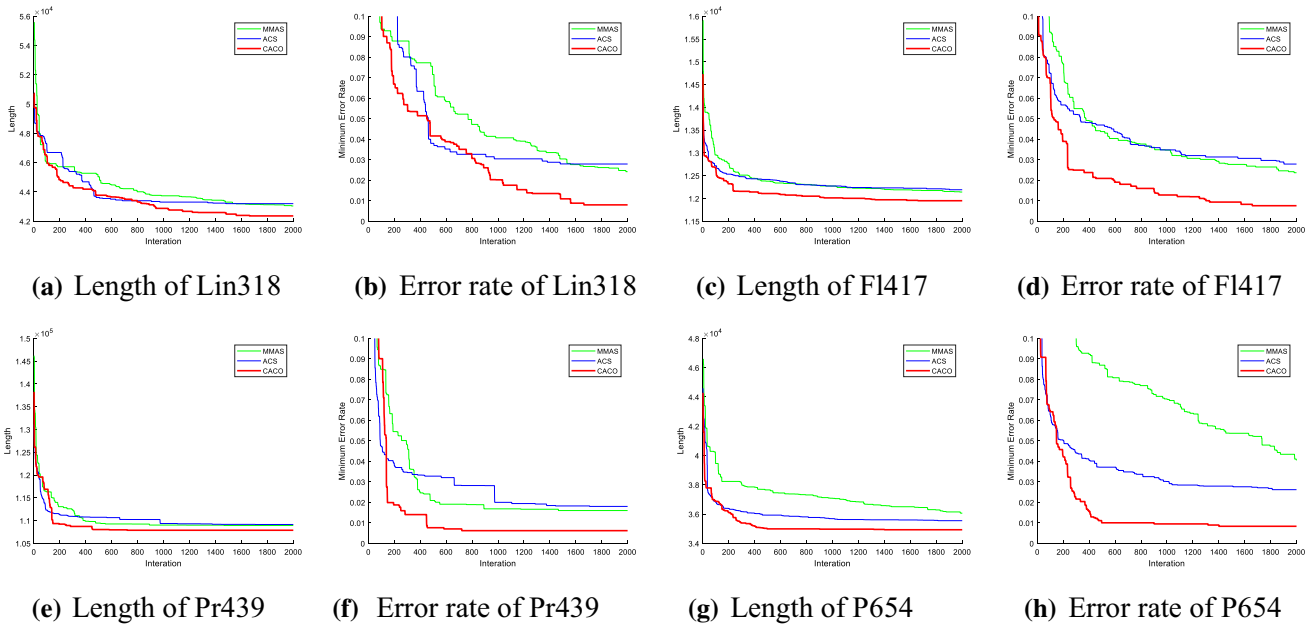


Fig. 10 Convergence comparison of MMAS, ACS and CACO

solution is greater than 4% starting from the large-scale city set Lin318. In contrast, CACO uses pheromone guidance for multiple populations, which makes the algorithm perform consistently on large-scale city sets and within 1% of the minimum error in the Pr439 city set, indicating that the accuracy of the solution in large-scale city sets is significantly better than MPSO.

The data in Table 4 are more comprehensive, comparing in terms of optimal solution, average solution, minimum error rate, and average error rate. It can be clearly seen that the performance of D-GWO is less stable in the small-scale city sets, and although it can find the approximate optimal solution, it still has some error. While CACO can find the optimal solution in the city sets compared with the number of cities within 150, and the average error rate is within 1%, so the performance is more stable. In larger city sets such as Pr226, although the minimum error rate and average error rate of D-GWO are small with better results, the error rate of the

optimal solution of CACO is only 0.004%, which is very close to the optimal solution. In the Pr439 city set, the performance of the D-GWO algorithm decreases significantly, but CACO is able to explore a better optimal route stably due to the guidance of pheromones after community detection, so the global search capability and stability are significantly better than D-GWO.

Overall, CACO has the ability to consistently explore optimal solutions for small sets of cities thanks to the small communities with high quality routes delineated by the route relationship network, allowing the algorithm to quickly find the optimum among high quality routes. At the same time, the pheromones of other routes in the small communities increase the chances that ants will choose other cities when guiding multiple populations, increasing the diversity of the algorithm.

For larger city sets, the exploration accuracy of CACO is high, with the error rate within an acceptable range, and the

Table 3 Comparison of the CACO with MPSO

Instance	TSP	Optimal	CACO		MPSO	
			Best	$E_{min}(\%)$	Best	$E_{min}(\%)$
Eil51		426	426	0.00	426	0.00
Pr76		108,159	108,159	0.00	108,159	0.00
Rat99		1211	1211	0.00	1211	0.00
KroA100		21,282	21,282	0.00	21,282	0.00
KroB100		22,141	22,141	0.00	22,141	0.00
Lin105		14,379	14,379	0.00	14,379	0.00
Pr107		44,303	44,303	0.00	44,303	0.00
Pr124		59,030	59,030	0.00	59,030	0.00
Ch130		6110	6110	0.00	6110	0.00
Pr144		58,537	58,537	0.00	58,537	0.00
Ch150		6528	6532	0.06	6528	0.00
KroA150		26,524	26,524	0.00	26,524	0.00
KroB150		26,130	26,130	0.00	26,254	0.47
KroA200		29,368	39,368	0.00	29,533	0.56
KroB200		29,437	29,524	0.30	29,501	0.22
A280		2579	2583	0.16	2598	0.74
Lin318		42,029	42,365	0.80	43,710	4.00
Pr439		107,217	107,883	0.62	111,875	4.34

Table 4 Comparison of the CACO with D-GWO

Instance	TSP	Optimal	CACO			D-GWO				
			Best	Avg	$E_{min}(\%)$	$E_{avg}(\%)$	Best	Avg	$E_{min}(\%)$	$E_{avg}(\%)$
Att48		33,522	33,522	33,548.35	0.00	0.08	33,523	33,600	0.003	0.23
Pr76		108,159	108,159	108,357.60	0.00	0.18	108,159	108,900	0.00	0.68
KroB100		22,141	22,141	22,235.00	0.00	0.42	22,159	22,444.6	0.08	1.37
KroC100		20,749	20,749	20,812.05	0.00	0.30	20,749	21,078	0.00	1.58
KroE100		22,068	22,068	22,190.05	0.00	0.55	22,131	22,410	0.29	1.54
Lin105		14,379	14,379	14,383.95	0.00	0.03	14,382	14,520	0.02	0.98
Pr107		44,303	44,303	44,566.70	0.00	0.60	44,303	44,685.1	0.00	0.86
Pr124		59,030	59,030	59,213.55	0.00	0.31	59,030	59,390.9	0.00	0.61
Pr144		58,537	58,537	58,600.00	0.00	0.11	58,537	58,600.5	0.00	0.11
KroB150		26,130	26,130	26,354.35	0.00	0.86	26,320	26,756.2	0.73	2.39
Pr226		80,369	80,373	80,940.90	0.004	0.71	80,648	81,135.7	0.35	0.95
Pr439		107,217	107,883	108,983.20	0.62	1.65	110,415	112,850.3	2.98	5.25

stability of the solution is strong without much fluctuation, which is mainly due to the great contribution of using community detection and pheromone guidance in the algorithm. The samples of community detection come from the route information of multiple population ants in different periods, so with the increase in iterations, the routes explored by ants are getting closer to the optimal solutions, and the community detection on this basis is beneficial to explore better routes.

4.4 Comparison with Other Improved Algorithms

To further verify the excellent performance of the CACO algorithm, the comparison results of CACO with Hybrid Max–Min ant system (HMMA) [41], Ant colony algorithm based on generalized Jaccard similarity(JCACO) [42], Spider Monkey Optimization(DSMO) [43], parallel cooperative hybrid ant colony optimization(PACO-3Opt) [44] and Heterogenous Adaptive Ant Colony Optimization(HAACO) [45] are added, as shown in Table 5.

Table 5 Comparison of the CACO with other improved algorithms

	TSP	Eil51	Eil76	Rat99	KroA100	KroA200	Tsp225	Lin318	Fl417	Pr439
CACO	OPT	426	538	1211	21,282	29,368	3916	42,029	11,861	107,217
	Best	426	538	1211	21,282	29,368	3923	42,365	11,950	107,883
	Average	427.1	539.75	1213.5	21,337.2	29,501.55	3978.3	42,878.45	12,018.2	108,683.2
	$E_{min}(\%)$	0.00	0.00	0.00	0.00	0.00	0.18	0.80	0.75	0.62
HMMA	Best	438.95	549.9	1255.39	21,371.36	29,999.54	4073.85	45,349.62	12,542.84	114,094.97
	Average	450.07	558.05	1303.78	21,425.9	31,405.69	4203.89	46,548.57	12,897.84	116,885.91
	$E_{min}(\%)$	3.04	2.21	3.67	0.42	2.15	4.03	7.90	5.75	6.41
JCACO	Best	426	538	1211	21,282	29,406	3935	42,399	11,969	108,375
	Average	427	539	1214	21,322	29,542	4007	43,168	12,216	110,700
	$E_{min}(\%)$	0.00	0.00	0.00	0.00	0.13	0.49	0.88	0.91	1.08
DSMO	Best	428.86	558.68	1225.56	21,298.21	30,481.35	4013.68	44,118.66	12,218.98	112,105.2
	Average	436.96	572.7	1291.93	22,024.27	31,828.64	4162.79	45,460.25	12,950.77	116,379.2
	$E_{min}(\%)$	0.67	3.84	1.20	0.08	3.79	2.49	4.97	3.02	4.56
PACO-3Opt	Best	426	538	1213	21,282	29,533	–	–	11,958	108,482
	Average	426.35	539.5	1217.1	21,326.8	29,646.05	–	–	11,991.9	108,702
	$E_{min}(\%)$	0.00	0.00	0.17	0.00	0.56	–	–	0.82	1.18
HAACO	Best	426	538	1211	21,282	29,483	–	–	–	–
	Average	427.5	542	1214.1	21,364.2	29,633.2	–	–	–	–
	$E_{min}(\%)$	0.00	0.00	0.00	0.00	0.39	–	–	–	–

From the number of optimal solutions obtained in different city sets, CACO is significantly better than the other.

algorithms. CACO, JCACO and HAACO can all find optimal solutions in city sets with less than 200 cities with good exploration ability. In the large-scale city set, CACO finds better solutions than other algorithms, and the minimum error rate is less than 1% even in Pr439. Comparing with the average solution, CACO generally has smaller average solutions than other algorithms in large-scale cities, and the stability of the algorithm is better when dealing with more cities.

4.5 Application of the Proposed Algorithm

The above experiments demonstrate the excellent performance of CACO in solving TSP problems, and we will play the advantages of CACO in combination with actual problems in the following. Those red points in Fig. 11 are the general hospitals in Songjiang District, Shanghai, China, located through ArcGIS and Google API.

In the context of COVID-19 affecting the world, assuming that here the medical resources need to be transported to all hospitals at once without interruption, this requirement can be modeled as a TSP problem, and Fig. 12 shows its community detection results and solution routes, with the total route length of 122.653 km and the distance of 40.762 km between A and B.

However, the relationship between community nodes in reality changes from time to time. When the situation of hos-

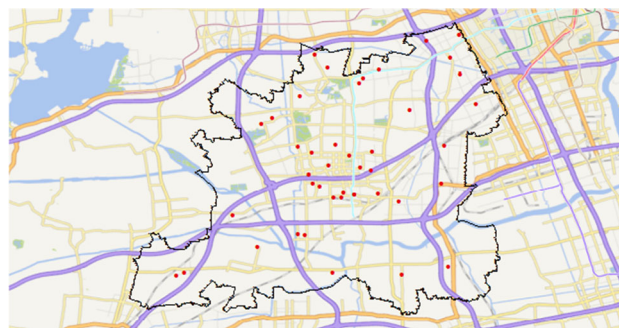


Fig. 11 Hospital location map

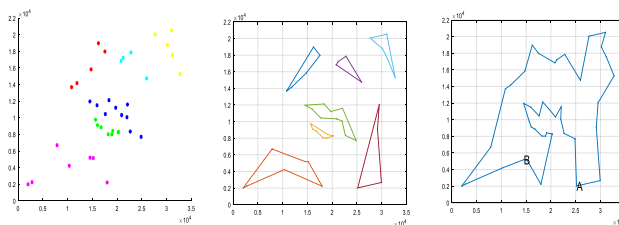


Fig. 12 The optimal results without considering A and B

pitals A and B is very urgent, priority should be given to transport resources to hospitals A and B, which cannot be carried out by the ordinary ant colony algorithm. Figure 13 shows the community detection results and solution routes after considering the situation of hospitals A and B. The total route length is 123.873 km, and the distance between A and

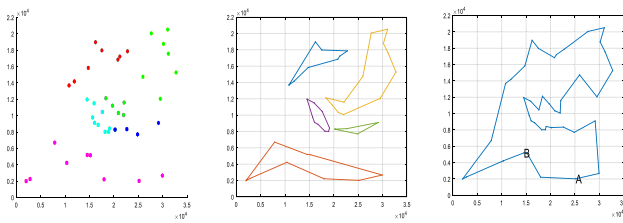


Fig. 13 The optimal results considering A and B

B is 11.897 km. Although the total length increases by 1%, the distance between A and B is shortened by 71%, which is an obvious improvement.

5 Conclusion and Future Direction

This study proposes the multiple ant colony algorithm combining community relationship network (CACO), which is different from most scholars who have improved around the route information of better ants. By incorporating the idea of community detection, this study sampled all the route information of different populations of ants to construct a route relationship network model belonging to multiple populations. In this model, community division is performed by modularity, and the small communities are combined with ant colony algorithm for solving the TSP problem. Through the processing of samples and community detection, the divided small communities are structurally stable and have the potential for optimal solutions. At this time, the pheromone of the small community is used to guide the multiple clusters, which speeds up the convergence speed. Also, the pheromones of non-optimal edges increase the diversity of multiple communities, balancing the speed of convergence and diversity.

CACO is compared with the traditional ant colony algorithm and the latest algorithms to date through 28 TSP city sets, and the simulation results show that CACO outperforms other algorithms, as well as better balance between convergence speed and diversity when dealing with large-scale problems.

For future research, we will perform more interdisciplinary combinations. Although ant colony algorithms incorporating community relationship networks can correlate distance with node sparsity, the literature that can be studied is relatively few and limited in thinking. So, there are more optimization studies in theory while relatively simple in application. In the future, more attention will be paid to the combination of complex network structures and intelligent algorithms to solve realistic and more complex problems.

References

- Johnson, D.S.; McGeoch, L.A.: The traveling salesman problem: Local Search Comb. Optim. **6**, 215–310 (2018). <https://doi.org/10.2307/j.ctv346t9c.13>
- Yoon, K.: Operational Research Society is collaborating with JSTOR to digitize, preserve, and extend access to Journal of the Operational Research Society. © www.jstor.org. J. Oper. Res. Soc. **38**, 277–286 (1987)
- Padberg, M.; Rinaldi, G.: Optimization of a 532-city symmetric traveling salesman problem by branch and cut. Oper. Res. Lett. **6**, 1–7 (1987). [https://doi.org/10.1016/0167-6377\(87\)90002-2](https://doi.org/10.1016/0167-6377(87)90002-2)
- Lü, X.; Wu, Y.; Lian, J.; Zhang, Y.; Chen, C.; Wang, P.; Meng, L.: Energy management of hybrid electric vehicles: a review of energy optimization of fuel cell hybrid power system based on genetic algorithm. Energy Convers. Manag. **205**, 112474 (2020). <https://doi.org/10.1016/j.enconman.2020.112474>
- Dereli, S.; Köker, R.: Strengthening the PSO algorithm with a new technique inspired by the golf game and solving the complex engineering problem. Complex Intell. Syst. **7**, 1515–1526 (2021). <https://doi.org/10.1007/s40747-021-00292-2>
- Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.: An improved grey wolf optimizer for solving engineering problems. Expert Syst. Appl. **166**, 113917 (2021). <https://doi.org/10.1016/j.eswa.2020.113917>
- Dereli, S.: A Novel approach based on average swarm intelligence to improve the whale optimization algorithm. Arab. J. Sci. Eng. (2021). <https://doi.org/10.1007/s13369-021-06042-3>
- Sundararaj, V.: Optimal task assignment in mobile cloud computing by queue based ant-bee algorithm. Wirel. Pers. Commun. **104**, 173–197 (2019). <https://doi.org/10.1007/s11277-018-6014-9>
- Yi, N.; Xu, J.; Yan, L.; Huang, L.: Task optimization and scheduling of distributed cyber–physical system based on improved ant colony algorithm. Futur. Gener. Comput. Syst. **109**, 134–148 (2020). <https://doi.org/10.1016/j.future.2020.03.051>
- Wang, J.; Cao, J.; Sherratt, R.S.; Park, J.H.: An improved ant colony optimization-based approach with mobile sink for wireless sensor networks. J. Supercomput. **74**, 6633–6645 (2018). <https://doi.org/10.1007/s11227-017-2115-6>
- Gao, S.; Wang, Y.; Cheng, J.; Inazumi, Y.; Tang, Z.: Ant colony optimization with clustering for solving the dynamic location routing problem. Appl. Math. Comput. **285**, 149–173 (2016). <https://doi.org/10.1016/j.amc.2016.03.035>
- Mu, C.; Zhang, J.; Liu, Y.; Qu, R.; Huang, T.: Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks. Soft Comput. **23**, 12683–12709 (2019). <https://doi.org/10.1007/s00500-019-03820-y>
- Shahabi Sani, N.; Manthouri, M.; Farivar, F.: A multi-objective ant colony optimization algorithm for community detection in complex networks. J. Ambient Intell. Humaniz. Comput. **11**, 5–21 (2020). <https://doi.org/10.1007/s12652-018-1159-7>
- Dorigo, M.; Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**, 53–66 (1997). <https://doi.org/10.1109/4235.585892>
- Stützle, T.; Hoos, H.H.: MAX-MIN Ant System. Futur. Gener. Comput. Syst. **16**, 889–914 (2000). [https://doi.org/10.1016/S0167-739X\(00\)00043-1](https://doi.org/10.1016/S0167-739X(00)00043-1)



16. Zhang, Q.; Zhang, C.: An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem. *Neural Comput. Appl.* **30**, 3209–3220 (2018). <https://doi.org/10.1007/s00521-017-2912-0>
17. Zhao, D.; Liu, L.; Yu, F.; Heidari, A.A.; Wang, M.; Oliva, D.; Muhammad, K.; Chen, H.: Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Syst. Appl.* **167**, 114122 (2021). <https://doi.org/10.1016/j.eswa.2020.114122>
18. Li, S.; Cai, S.; Li, L.; Sun, R.; Yuan, G.: CAAS: a novel collective action-based ant system algorithm for solving TSP problem. *Soft Comput.* **24**, 9257–9278 (2020). <https://doi.org/10.1007/s00500-019-04452-y>
19. Wu, Y.; Ma, W.; Miao, Q.; Wang, S.: Multimodal continuous ant colony optimization for multisensor remote sensing image registration with local search. *Swarm Evol. Comput.* **47**, 89–95 (2019). <https://doi.org/10.1016/j.swevo.2017.07.004>
20. Tam, J.H.; Ong, Z.C.; Ismail, Z.; Ang, B.C.; Khoo, S.Y.: A new hybrid GA–ACO–PSO algorithm for solving various engineering design problems. *Int. J. Comput. Math.* **96**, 883–919 (2019). <https://doi.org/10.1080/00207160.2018.1463438>
21. Gülcü, Ş.; Mahi, M.; Baykan, Ö.K.; Kodaz, H.: A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. *Soft Comput.* **22**, 1669–1685 (2018). <https://doi.org/10.1007/s00500-016-2432-3>
22. Abdelbar, A.M.; Salama, K.M.: Parameter self-adaptation in an ant colony algorithm for continuous optimization. *IEEE Access.* **7**, 18464–18479 (2019). <https://doi.org/10.1109/ACCESS.2019.2896104>
23. Castillo, O.; Neyoy, H.; Soria, J.; Melin, P.; Valdez, F.: A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot. *Appl. Soft Comput. J.* **28**, 150–159 (2015). <https://doi.org/10.1016/j.asoc.2014.12.002>
24. Wang, M.; Ma, T.; Li, G.; Zhai, X.; Qiao, S.: Ant colony optimization with an improved pheromone model for solving MTSP with capacity and time window constraint. *IEEE Access.* **8**, 106872–106879 (2020). <https://doi.org/10.1109/ACCESS.2020.3000501>
25. Deng, W.; Xu, J.; Song, Y.; Zhao, H.: An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application. *Int. J. Bio-Inspired Comput.* **16**, 158–170 (2020). <https://doi.org/10.1504/IJBIC.2020.111267>
26. Deng, W.; Zhao, H.; Zou, L.; Li, G.; Yang, X.; Wu, D.: A novel collaborative optimization algorithm in solving complex optimization problems. *Soft Comput.* **21**, 4387–4398 (2017). <https://doi.org/10.1007/s00500-016-2071-8>
27. Dai, X.; Long, S.; Zhang, Z.; Gong, D.: Mobile robot path planning based on ant colony algorithm with a* heuristic method. *Front. Neurobot.* (2019). <https://doi.org/10.3389/fnbot.2019.00015>
28. Mohsen, A.M.: Annealing ant colony optimization with mutation operator for solving TSP. *Comput. Intell. Neurosci.* (2016). <https://doi.org/10.1155/2016/8932896>
29. Li, S.; You, X.; Liu, S.: Multiple ant colony optimization using both novel LSTM network and adaptive Tanimoto communication strategy. *Appl. Intell.*, (2021). <https://doi.org/10.1007/s10489-020-02099-z>
30. Gong, Y.; Gu, T.; Zhao, F.; Yuan, H.: Multiobjective cloud workflow scheduling : approach. *IEEE Trans. Cybern.* **49**, 2912–2926 (2019)
31. Zhang, Z.; Yang, W.; Li, J.: Image feature extraction based multiple ant colonies cooperation. *Autom. Target Recognit.* **XXV**, 9476, 947615 (2015). <https://doi.org/10.1117/12.2176542>
32. Xu, M.; You, X.; Liu, S.: A novel heuristic communication heterogeneous dual population ant colony optimization algorithm. *IEEE Access.* **5**, 18506–18515 (2017). <https://doi.org/10.1109/ACCESS.2017.2746569>
33. Mehrle, D.; Strosser, A.; Harkin, A.: Walk-modularity and community structure in networks. *Netw. Sci.* **3**, 348–360 (2015). <https://doi.org/10.1017/nws.2015.20>
34. Newman, M.E.J.; Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E- Stat. Nonlinear, Soft Matter Phys.* **69**, 1–16 (2004). <https://doi.org/10.1103/PhysRevE.69.026113>
35. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, 1–12 (2008). <https://doi.org/10.1088/1742-5468/2008/10/P10008>
36. Noveiri, E.; Naderan, M.; Alavi, S.E.: Community detection in social networks using ant colony algorithm and fuzzy clustering. 2015 5th Int. Conf. Comput. Knowl. Eng. ICCKE 2015. 73–79 (2015). doi: <https://doi.org/10.1109/ICCKE.2015.7365864>
37. Ben Romdhane, L.; Chaabani, Y.; Zardi, H.: A robust ant colony optimization-based algorithm for community mining in large scale oriented social graphs. *Expert Syst. Appl.* **40**, 5709–5718 (2013). <https://doi.org/10.1016/j.eswa.2013.04.021>
38. Lancichinetti, A.; Fortunato, S.; Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E- Stat. Nonlinear, Soft Matter Phys.* **78**, 1–5 (2008). <https://doi.org/10.1103/PhysRevE.78.046110>
39. Yousefikhoshbakht, M.: Solving the traveling salesman problem: a modified metaheuristic algorithm. *Complexity.* (2021). <https://doi.org/10.1155/2021/6668345>
40. Panwar, K.; Deep, K.: Discrete grey wolf optimizer for symmetric travelling salesman problem. *Appl. Soft Comput.* **105**, 107298 (2021). <https://doi.org/10.1016/j.asoc.2021.107298>
41. Yong, W.: Hybrid Max-Min ant system with four vertices and three lines inequality for traveling salesman problem. *Soft Comput.* **19**, 585–596 (2015). <https://doi.org/10.1007/s00500-014-1279-8>
42. Zhang, D.; You, X.; Liu, S.; Yang, K.: Multi-Colony Ant Colony Optimization Based on Generalized Jaccard Similarity Recommendation Strategy. *IEEE Access.* **7**, 157303–157317 (2019). <https://doi.org/10.1109/ACCESS.2019.2949860>
43. Akhand, M.A.H.; Ayon, S.I.; Shahriyar, S.A.; Siddique, N.; Adeli, H.: Discrete spider monkey optimization for travelling salesman problem. *Appl. Soft Comput. J.* **86**, 105887 (2020). <https://doi.org/10.1016/j.asoc.2019.105887>
44. Mahi, M.; Baykan, Ö.K.; Kodaz, H.: A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. *Appl. Soft Comput. J.* **30**, 484–490 (2015). <https://doi.org/10.1016/j.asoc.2015.01.068>
45. Tuani, A.F.; Keedwell, E.; Collett, M.: heterogeneous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. *Appl. Soft Comput.* **97**, 106720 (2020). <https://doi.org/10.1016/j.asoc.2020.106720>