Applied Intelligence (2021) 51:752–774
https://doi.org/10.1007/s10489-020-01841-x

Pearson correlation coefficient-based pheromone refactoring mechanism for multi-colony ant colony optimization

Han Pan[1] · Xiaoming You[1] · Sheng Liu[1] · Dehui Zhang[1]

Published online: 29 August 2020

Abstract

To solve the problem of falling into local optimum and poor convergence speed in large Traveling Salesman Problem (TSP), this paper proposes a Pearson correlation coefficient-based Pheromone refactoring mechanism for multi-colony Ant Colony Optimization (PPACO). First, the dynamic guidance mechanism is introduced to dynamically adjust the pheromone concentration on the path of the maximum and minimum spanning tree, which can effectively balance the diversity and convergence of the algorithm. Secondly, the frequency of communication between colonies is adjusted adaptively according to a criterion based on the similarity between the minimum spanning tree and the optimal solution. Besides, the pheromone matrix of the colony is reconstructed according to the Pearson correlation coefficient or information entropy to help the algorithm jump out of the local optimum, thus improving the accuracy of the solution. These strategies greatly improve the adaptability of the algorithm and ensure the effectiveness of the interaction. Finally, the experimental results indicate that the proposed algorithm could improve the solution accuracy and accelerate the convergence speed, especially for large-scale TSP instances.

Keywords Pearson correlation coefficient · Multi-colony · TSP · Information entropy · Minimum spanning tree

1 Introduction

Traveling Salesman Problem is a typical combinatorial optimization problem and a Non-deterministic Polynomial Complete (NP-C) problem. It is a concentrated generalization and simplified form of various complex problems in many fields and has become an indirect comparison standard of heuristic search and optimization algorithms. Besides, TSP is widely used in many fields, such as transportation, computer network, circuit board design, and logistics distribution. Therefore, it has important theoretical value and high practical value to solve TSP quickly and effectively. The TSP can be explained as that the traveler knows the mutual distances between n cities. He starts from a certain city, then visits each city once and only once before returning to the first city. Lastly, he asks to find the shortest path to traverse n cities. In the beginning, optimal solution algorithms have been used to solve TSP, such as the branch and bound method and the dynamic programming method. Although the optimal solution algorithm yields exact solution, the computation time is intolerable and hence various approximation methods have been developed, such as the MM algorithm, greedy algorithm, and the MST algorithm. These approximation algorithms are not satisfactorily close to the optimal solution, although they can obtain feasible solutions that are close to the optimal solution relatively quickly.

Subsequently, various meta-heuristic algorithms were proposed to solve TSP, which are a combination of stochastic algorithms and local search algorithms. They usually do not rely on conditions specific to certain problems and thus can be applied to a wider range of aspects. Today, meta-heuristic algorithms have been successfully applied in engineering, computer network, biological system modeling, forecasting, pattern recognition, data clustering, and feature selection, etc [1–4]. Meta-heuristic algorithms are classified into local search-based algorithms and population-based algorithms. Although local search algorithms are simple, flexible, and easy to implement, they tend to fall into the local optimum, such as simulated annealing [5],

✉ Xiaoming You
yxm6301@163.com

Han Pan
panhansh@126.com

[1] Shanghai University of Engineering Science, Shanghai, China

tabu search [6], hill climbing [7], etc. Population-based methods can be divided into evolutionary computation and swarm intelligence methods. Evolutionary computation is a well-established global optimization method with high robustness and wide applicability, such as Genetic Algorithm (GA) [8], Evolutionary Strategy (ES) [9], Evolutionary Programming (EG) [10]. However, the algorithms are dependent on the choice of parameters and most importantly on their poor local search capability. Swarm intelligence algorithms primarily simulate the behavior of groups of insects, herds, birds, and fish. These groups follow a cooperative approach to finding food, with each member of the group constantly changing the direction of the search by learning from their own experiences and the experiences of other members. It is a bionic and stochastic search algorithm with robustness and intelligence, such as Ant Colony Optimization (ACO) [11], Particle Swarm Optimization (PSO) [12], Artificial Bee Colony (ABC) [13], and Fruit Fly Optimization Algorithm(FOA) [14]. The swarm intelligence algorithm enables the exchange of information and mutual collaboration between individuals and groups. The individual has a certain degree of randomness, which maintains the diversity of search directions to some extent and prevents premature convergence. The group grasps the direction of optimization on the whole, thus ensuring the convergence of the algorithm. Therefore, in recent years, more and more swarm intelligence algorithms have been used to solve TSP [15–18].

The ant colony optimization is a probabilistic algorithm for finding optimal paths. It was proposed by M. Dorigo in his doctoral thesis in 1992 [19], and it was inspired by the behavior of ants discovering paths in the process of finding food. As ants walk,they release a substance called pheromones that are used to mark their walking path. In the process of searching for food, the ants choose the direction of walking according to pheromone concentration and eventually reach the food. Compared with other heuristics, the ant colony algorithm is characterized by distributed computing, pheromone positive feedback, and strong robustness. As a result, ant colony optimization has been widely used in Recommender systems [20], Feature selection [21], machine layout problem [22], path planning problem [23], and other fields, and has obtained remarkable results. Especially for TSP, many experts and scholars have solved it better by improving ant colony optimization. However, the multi-colony algorithm is superior to the single colony algorithm in performance. Therefore, the research of the multi-colony algorithm has become an inevitable trend. In the recommendation system, Pearson correlation coefficient is used to measure the similarity between two user interests. And in the multi-colony algorithm, there are similar parts between the paths found by different sub-colonies, so some scholars combine Pearson correlation coefficient with multi-colony algorithms as a way to improve the performance of the algorithm [24].

Inspired by the above analysis, a Pearson correlation coefficient-based pher-omone refactoring mechanism for multi-colony ant colony optimization is proposed in this paper. We focus on improving the accuracy of the solution on large-scale TSPs and balance the diversity and the convergence performance of the algorithm. We selected 13 TSP instances of different scales to verify the performance of PPACO and compared them with the traditional ant colony algorithms and the improved metaheuristic algorithms in this field. Second, in order to illustrate that PPACO is different from the traditional ant colony algorithms and improved ant colony algorithms, we use Friedman to prove the statistical significance of the results. Aside from this, in order to obtain as superior a set of parameters as possible, we use the orthogonal test to select the appropriate parameters. Finally, the experimental results show that the performance of PPACO is better than several algorithms mentioned in this paper. It can effectively accelerate the convergence speed while alse obtaining more accurate solutions. The main contributions of this paper are summarized as follows:

1. Ant colony optimization based on a dynamic guidance mechanism (DGMACO) is proposed as a novel single colony. It dynamically adjusts the phero-mone concentration on the path of the maximum and minimum spanning tree, which can effectively balance the diversity and convergence of the algorithm.
2. The similarity between the minimum spanning tree and the optimal path is used as the evaluation criterion to adaptively adjust the communication frequency of the colony, thus helping the algorithm to jump out of the local optimum.
3. According to the Pearson correlation coefficient or information entropy, the pheromone matrix of sub-colonies can be reconstructed so that a high-precision solution can be obtained.

This article is organized as follows: Section 2 introduces the related works in the domain of the ACO and the motivation of our work. Section 3 describes the ACS, MMAS, Maximum spanning tree and Minimum spanning tree, Pearson correlation coefficient, and Information entropy. The dynamic guidance mechanism and pheromone refactoring mechanism are proposed in Section 4. Section 5 illustrates the experimental results of TSPs and the comparison between different algorithms. Section 6 summarizes our work and describes some of our future directions.

2 The related work

In this section, we briefly review some research work in related areas and discuss the differences and connections between these works and our method.

Ant Colony Optimization is one of the most effective metaheuristic algorithms, which simulates the foraging behavior of ants in nature. It has been successfully applied to solve combinatorial optimization problems. In 1992, M.Dorigo et al. proposed the Ant System (AS) [25] inspired by the mechanism of biological evolution. Since AS has the characteristics of positive information feedback, distributed computing, and heuristic search, it has been widely concerned and studied by many scholars. However, with the increasing scale of test cases, the performance degrades severely. The main drawback is the slow rate of convergence and the tendency to fall into local optimum.

To solve these problems, M. Dorigo proposed the ant colony system (ACS) in 1996 [26], which is a modified algorithm based on the AS. The algorithm introduces the concept of a global update, and the state transition rules used in path creation are also superior to the AS. As a result, ACS can get a better solution when solving large TSP instances. In 1997, T.S tutzle et al., in the experimental analysis and application research of AS, proposed the Max-Min Ant System (MMAS) [27]. In the MMAS, each iteration allows only the best ant to update the pheromone path, and limits the upper and lower values of pheromone concentration. The purpose of this is to prevent premature stagnation of the algorithm, and increase the diversity of the algorithm. However, the mentioned traditional ant colony algorithms still have some defects, such as insufficient convergence, low precision, and easy to fall into local optimum.

To enhance the global search ability and increase the diversity of the algorithm, a large number of variations of ACO have been presented over the past few years. Wu et al. proposed a multimodal continuous ant colony optimization algorithm and designs an efficient local search optimization method to ensure high diversity and improve search efficiency [28]. Ratanavilisagul applied mutation operation to pheromone to improve the diversity of ant colony [29]. Gao introduces a premium-penalty strategy to polarize the pheromone density of all paths [30]. Ye et al. proposed using search-history information to continuously acquire failure experience to guide the ant colony to explore unknown space during the optimization process, and to utilize the negative feedback to improve the diversity of solutions [31]. Chen et al. proposed a method to adjust the time interval adaptively according to the diversity of the solutions, to increase the ability of the search and avoid early convergence [32].

The above improvements of ACO have greatly boosted the performance. But the convergence speed of the algorithm still needs to be improved. For the above deficiencies, a novel strengthened pheromone updating mechanism is designed, which strengthens pheromone on the edge that never appeared before, using the dynamic information in the process of the optimal path optimization, to achieve the purpose of strengthening the convergence speed [33]. The main idea of Shetty et al. is to generate multiple minion ants to help the parent ant make a better decision. Besides, they control the number of minion ants created by each parent ant to decrease futile traversal of the same paths again and again [34]. Luo et al. proposed the inverse feedback mechanism for the dissemination of pheromones and the positive feedback mechanism for pheromone concentration to accelerate the convergence speed [22]. Jun-man et al. designed a novel optimized implementation approach to enhance the impact of pheromones on ants, which can accelerate the convergence of the algorithm [35].

To keep a more reasonable balance between the search ability and the convergence in the search process, some scholars have proposed a hybrid algorithm based on ant colony algorithm. The hybrid algorithm based on ant colony algorithm can absorb the advantages of other algorithms so as to obtain better performance [36–39]. Luan et al. proposed a hybrid algorithm of genetic algorithm and ant colony optimization, which combines the merits of GA with great global converging rate and ACO with parallelism and effective feedback, so the diversity and convergence of the algorithm are improved [40]. In order to effectively balance diversity and convergence, Mohsen et al. integrated the advantages of ACO, SA, mutation operator, and local search procedure to solve the traveling salesman problem [41]. To overcome the slow convergence of the ant colony optimization algorithm for continuous domain problems, a novel hybrid algorithm based on ant colony optimization and particle swarm optimization algorithm was proposed by Zhang et al. [42]. Mahi et al. proposed a hybrid algorithm based on particle swarm optimization, ant colony optimization, and 3-opt, which enhanced the ability of the algorithm to jump out of local optimization and improve the quality of the solution [43].

However, all the above algorithms are single colony ant colony algorithms. To further improve the search performance and solution quality of the ant colony algorithm, the multi-colony ant colony algorithms have been proposed [44–47]. Different ant colonies have different characteristics, complementary advantages, and potential cooperation with each other, so heterogeneous multi-colony ant colony algorithms have more advantages in solving complex and large-scale problems. Tuani et al. solved hard Optimization problems by introducing unique biases towards the pheromone trail and local heuristics for each ant. Besides, the well-known Ant System and

Max-Min Ant System are used as the base algorithms to implement heterogeneity, so as to effectively improve the quality of the solution [48]. A heterogeneous feature ant colony optimization algorithm based on effective vertexes of obstacles is proposed by Zhao et al. to solve the problem of poor convergence and local optimum [49]. Lee et al. proposed a heterogeneous-ants-based path planner for global path planning of mobile robot applications so that it could directly find an optimal and smoother path without post-processing to smooth the path [23]. Yao et al. proposed a heterogeneous feature ant colony optimization algorithm to solve the robot path planning problem. In the proposed method, two kinds of ants with different features are designed to influence the convergence rate of the algorithm by controlling the number of them [50].

The inter-species communication of multi-colony ant colony algorithm allows information sharing among the colonies, so as to effectively adjust their own search experience and reduce search time. Therefore, the selection of the communication period is one of the important issues to be considered in a multi-colony algorithm. Xu et al. pointed out that communication among sub-colonies every other constant period can effectively prevent the colony from falling into the local optimum [51]. However, the choice of this fixed constant is not easy. If the selected constant is too small, the communication will be too frequent, which will disturb the internal search experience of the sub-colony and increase the computation time. If the constant is too large, the timely information sharing among the colonies is not achieved, which will reduce the search performance of the colony and increase the probability of stagnation. We found that the adaptive communication cycle can better serve the colony. It allows the colony to adaptively adjust the frequency of communication according to their search conditions and progress. Besides, the selection of a multi-colony interaction strategy is also crucial for the performance of the algorithm. However, the interaction strategies of some existing multi-colony algorithms are relatively simple, and the adaptability needs to be improved. To solve this problem, the Pearson correlation coefficient is introduced as the evaluation criterion to reward the parameters of similar paths with an adaptive frequency [24]. Thus, the probability of the colony falling into the local optimum is reduced. Pearson correlation coefficient is to examine the degree of correlation between two things. When the Pearson correlation coefficient is applied to the ant colony algorithm, we can predict the similarity between sub-colonies, so we can better carry out inter-species interaction.

For better performance, a colony owning both of the above advantages should also be applied. In addition, the selection of a single population is also important. We select two classical ant colony algorithms ACS, MMAS and a new ant colony algorithm DGMACO to compose the multi-colony algorithm. Among them, ACS is a representative single colony ant colony algorithm in terms of convergence, and MMAS is a representative single colony ant colony algorithm in terms of diversity. DGMACO is the single colony ant colony algorithm proposed in this paper, which is responsible for balancing diversity and convergence. The combination of these single colonies with different characteristics can improve the performance of the algorithm. All of the above are what motivate our work.

The main advantages and disadvantages of the previously published improved ACO are reported in Table 1. The methods are evaluated based on their general realizations and evaluations as reported in the literature. The numbers in the table represent the algorithm corresponding to the literature.

3 Materials and methods

3.1 The principle of ACS

3.1.1 Path construction

To solve the defects shown in AS, the ACS was proposed, which is an optimization algorithm based on Ant-Q. In the ACS, the selection mechanism from city i to city j is based on pseudo-random proportion rule. The rule is as follows:

$$j = \begin{cases} \arg \max\limits_{j \in allowed} \{\tau_{ij} \cdot \eta_{ij}^{\beta}\} & q \le q_0 \\ J & else \end{cases} \tag{1}$$

Where q is a random variable uniformly distributed in the interval [0, 1]. $q_0 (0 \le q_0 \le 1)$ is a parameter, η_{ij} is the reciprocal of the distance between city i and city j, τ_{ij} represents the total number of pheromones between city i and city j, *allowed* means a collection of cities not on the ant taboo list. J is a random variable generated according to the probability distribution given by (2):

$$P_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{s \in allowed} [\tau_{is}(t)]^{\alpha}[\eta_{is}]^{\beta}} & j \in allowed \\ 0 & else \end{cases} \tag{2}$$

Where α and β respectively represent the weight of pheromone and heuristic factor in probability calculation.

3.1.2 Pheromone updates

1. Global pheromone update: Pheromone update is applied to the edges on the current best path, and the updated rule is shown in (3):

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

Table 1 Analyze the advantages and disadvantages of the improved ant colony algorithm

Improved ACO	Type of algorithm	advantages	disadvantages
[28–32]	Single colony ACO	High diversity and good search ability	Slow convergence
[22, 33–35]	Single colony ACO	Strong convergence	The precision of the solution is not high
[36–43]	Hybrid algorithm based on ACO	A more reasonable balance between the search ability and the convergence, prevent stagnation	Limited performance of single colony
[23, 24, 44–51]	Multi-colony ACO	high-quality solutions in complex and large-scale problems, fast convergence	Inadequate adaptability

Where $\Delta\tau_{ij} = 1/L_{gb}$, L_{gb} represents the length of the optimal path so far, $\Delta\tau_{ij}$ is the amount of pheromone increase and ρ represents the evaporation rate of the global pheromone. The update of global pheromone is helpful to improve the convergence speed of the algorithm.

2. Local pheromone update: When ant k passes the next city j from city i, the updating rule of pheromone on the path between city i and city j is shown in (4):

$$\tau_{\mathrm{ij}}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_0 \tag{4}$$

Where ρ represents the evaporation rate of local pheromone and τ_0 represents the initial value of the pheromone on each path. The update of local pheromone results in a corresponding reduction of pheromones at each edge, which prevents the algorithm from falling into local optimization prematurely and increases the diversity of the algorithm.

3.2 Max-Min ant system

To make the algorithm search near the shortest path and gradually find the global optimal solution, the algorithm only updates the pheromone of the shortest path in the current cycle. The formula is as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{5}$$

To prevent some edge pheromones from growing too fast and causing stagnation, the size of any edge pheromone in the MMAS algorithm is limited to the range of $[\tau_{\min}, \tau_{\min}]$. If the concentration of pheromone on the current edge is higher than $\tau_{\max}$, then the concentration of pheromone on the current edge is set to $\tau_{\max}$, as shown in the following formula:

$$\tau_{\max} = \frac{1}{1-\rho}\frac{1}{f(s^{opt})} \tag{6}$$

Where $f(s^{opt})$ is the global optimal solution and ρ is the volatile factor of pheromone.

If the concentration of the pheromone on the current edge is lower than $\tau_{\min}$, then the concentration of the pheromone on the edge is set to $\tau_{\min}$, as shown in the following formula:

$$\tau_{\min} = \frac{\tau_{\max}(1-\sqrt[n]{p_{best}})}{(n/2-1)\sqrt[n]{p_{best}}} \tag{7}$$

Where, p_{best} is the probability of finding the optimal solution when the MMAS algorithm converges, which is generally 0.05.

3.3 Maximum spanning tree and minimum spanning tree

3.3.1 Definition of maximum and minimum spanning trees

In a completely connected graph $G = (V, E)$ for given n vertices, the edge between vertex u and vertex v is expressed as (u, v), and $w(u, v)$ represents the weight of this edge. If there is a graph where T is a subset of E and is non-cyclic so that $w(T)$ is minimum, then T is the minimum spanning tree of G; otherwise, if $w(T)$ is maximum, then T is the maximum spanning tree of G. $w(T)$ is shown in the following formula:

$$w(T) = \sum_{(u,v)\in T} w(u, v) \tag{8}$$

3.3.2 The construction of maximum and minimum spanning trees

The construction of maximum and minimum spanning trees has many famous methods, such as the Kruskal algorithm and the Prim algorithm.

1. Main ideas of Kruskal algorithm: First, the n vertices of G are regarded as n isolated connected branches and all edges are sorted by weight from small to large. Then, edge weights are increased in order. If there is a circle

after adding an edge, the edge is not added, and the finally connected graph is formed.

2. Main ideas of Prim algorithm: Given the connected graph G and any root node r, the minimum spanning tree grows from node r until it covers all the nodes in V. In other words, the minimum spanning tree keeps searching for lightweight edges to achieve the sum of the minimum weights.

3.4 Pearson correlation coefficient

The user-based collaborative filtering algorithm finds neighboring users based on their preferences for items, and then the favorite items of the neighboring users are recommended to the current users. In the calculation, the user's preference for all items is taken as a vector to calculate the similarity between users. After finding K neighbors, according to the similarity weight of neighbors and their preference for items, the items not involved in the current user's preference are predicted, and a sorted list of items is calculated as a recommendation. At present, Cosine similarity, Modified cosine similarity, and Pearson similarity (9) are mainly used to calculate similarity $sim(u, v)$ in the collaborative filtering algorithm.

$$sim(u, v) = \frac{\sum\limits_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum\limits_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \tag{9}$$

Where, $I_{u,v}$ represents the set of graded items common to users u and v, $r_{u,i}$ represents a score of user u for an item i, and $r_{v,i}$ represents a score of user v for an item i. $\bar{r}_u$ and $\bar{r}_v$ respectively represent the average score of user u and user v on the project.

Since the Pearson correlation coefficient takes into account the users' rating scale and the grading matrix in this paper is a dense matrix, which can accurately calculate the similarity between users, the Pearson correlation coefficient is selected as the basis of similarity in this paper. The mathematical symbols in (9) correspond to the meanings of the algorithm in this paper: $I_{u,v}$ represents the common path in the optimal solution between the sub-colony u and the sub-colony v. $r_{u,i}$ represents the pheromone concentration of sub-colony u on path i. $r_{v,i}$ represents the pheromone concentration of sub-colony v on path i. $\bar{r}_u$ represents the mean value of the pheromone concentration of the unrepeated path in the pheromone matrix of sub-colony u. $\bar{r}_v$ represents the mean value of the pheromone concentration of the unrepeated path in the pheromone matrix of sub-colony v.

3.5 The information entropy

Information entropy is often used as a quantitative index of the information content of a system, which can be further used as the objective of system equation optimization or the criterion of parameter selection. If the system is more complex and there are more kinds of different situations, its information entropy is relatively large. If a system is simpler, there are fewer kinds of situations and less information entropy. Definition formula of information entropy:

$$H(X) = -\sum_{x \in X} P(x) \log(P(x)) \tag{10}$$

Where x is a solution, and X is the set of all the solutions. *P(x)* is the probability of the solution x. In the ant colony algorithm, the higher the information entropy of each iteration, the better the diversity of solutions.

4 Pearson correlation coefficient-based pheromone refactoring mechanism for multi-colony ant colony optimization

4.1 Dynamic guidance mechanism

We introduce the dynamic guidance mechanism into the ACS and propose a new single colony ant colony optimization named DGMACO. Moreover, it forms a multi-colony together with ACS and MMAS, which can further improve the overall search performance of the colony.

In solving the TSPs, the minimum spanning tree is closely related to the standard optimal path. In general, 70% $\sim$ 80% [52] of the paths in the minimum spanning tree are consistent with the optimal path of the TSP. Inspired by this, the information of the minimum spanning tree is combined to solve the TSP. More specifically, the pheromones on the path of the minimum spanning tree of each iteration are dynamically changed to speed up the convergence speed of the algorithm in the process of solving the problem. Therefore, this method plays a role of dynamic positive guidance.

In the TSP instances, ants choose the next city based on two factors, one is the pheromone concentration heuristic information between city i and city j, and the other is the distance heuristic information between city i and city j, namely the reciprocal of the distance between city i and city j. The initial pheromone between cities is a constant. At the first iteration of the algorithm, the pheromone on each path is equally attractive to the ant, but the distance heuristic information between the cities is different. So at the beginning of the algorithm, the probability of ant choosing the next city is dominated by distance heuristic information. Therefore, the shorter the distance between cities, the higher the probability of choosing the path. However, the solution of this path is not necessarily the optimal solution, so it is easy to fall into the local optimum. To avoid this defect, the pheromone of n-1 edges of the maximum spanning tree will be dynamically strengthened in the early stage of the algorithm, so as to increase the diversity of the algorithm and play a role of dynamic negative guidance.

To prevent excessive accumulation of pheromones, the upper and lower limits of the pheromone mechanism of MMAS are utilized. Besides, under the common influence of maximum and minimum spanning trees, the algorithm can effectively balance the diversity and convergence of the algorithm.

4.1.1 Minimum spanning tree strategy

The pheromone released according to the dynamic guidance mechanism is called the spanning-tree pheromone. In the early stage of the algorithm, the information of the minimum spanning tree is combined. When the current number of iterations is $n \in [1, n_0]$, spanning-tree pheromones are dynamically changed on the path of the minimum spanning tree, and the amount of spanning-tree pheromones is as follows:

$$\tau_{tree}^{s} = \rho_s \tau_0 \times \frac{n}{N} \tag{11}$$

Where N is the maximum number of iterations, τ_0 is the initial value of edge pheromone, ρ_s represents the minimum spanning tree adjustment factor, which is used to adjust the influence weight of spanning-tree pheromones in the early stage of the algorithm. Compared with the results of multiple experiments, when n_0 is taken as [N/3] and ρ_s is taken as 1, the experimental results in this paper are the best. The selection method of experimental parameters is given in Section 5. To obtain strong convergence ability, τ_{tree}^{s} is designed as an increasing function so that the algorithm can achieve rapid convergence.

4.1.2 Maximum spanning tree strategy

In this subsection, the information of the maximum spanning tree is combined in the early stage of the algorithm. When the current iteration number is $n \in [1, n_0]$, the algorithm dynamically releases a certain amount of spanning-tree pheromones on the path of the maximum spanning tree. The amount of spanning-tree pheromones is as follows:

$$\tau_{tree}^{b} = \rho_b \tau_0 \times \frac{N - n}{N} \tag{12}$$

Where N is the maximum number of iterations, τ_0 is the initial value of edge pheromone, and ρ_b is the maximum spanning tree adjustment factor, which is used to adjust the influence weight of spanning-tree pheromone in the early stage of the algorithm. Compared with the results of multiple experiments, when n_0 is taken as [N/3] and ρ_b is taken as 1, the experimental results in this paper are the best. The selection method of experimental parameters is given in Section 5. To improve the search ability while keeping fast convergence, τ_{tree}^{b} is designed as a minus function. Therefore, the increment of spanning-tree pheromones in the maximum spanning tree path decreases in each iteration, so that they do not accumulate excessively. As a consequence, the maximum spanning tree strategy can improve the diversity of the algorithm in the early stage and effectively prevent stagnation.

Because of the above analysis, we know that the minimum spanning tree strategy can accelerate the convergence rate and the maximum spanning tree strategy can improve diversity. These two strategies work together and complement each other, which can effectively achieve the goal of balancing convergence and search ability.

4.2 Pheromone refactoring mechanism

4.2.1 Adaptive interaction frequency

ACS, MMAS, and DGMACO are selected to form a multi-colony ant colony algorithm in this paper, in which each sub-colony has a different regulation mechanism of pheromone, thus improving the search space and search efficiency. However, if sub-colonies do not communicate with each other and only rely on their single pheromone update mechanism, it is still easy to fall into local optimization, especially for large TSP instances. Meanwhile, frequent interactions between sub-colonies can lead to an excessive accumulation of pheromones and high time complexity. Furthermore, the fixed periodic exchange between the sub-colonies is not conducive to the adaptive development of the sub-colonies, so that the advantages of each sub-colony cannot fully complement each other. Therefore, the choice of the communication cycle is very important. An adaptive interaction frequency is proposed in this paper, which helps sub-colonies to adaptively adjust the frequency of communication with other sub-colonies according to their search conditions.

70% $\sim$ 80% of the paths in the minimum spanning tree are consistent with the optimal path of the TSP. Therefore, when the similarity between the minimum spanning tree path and the optimal path of the sub-colony reaches a certain threshold, and if the similarity value remains unchanged for a period of iteration, the sub-colony can be considered to be immersed in local optimum. These sub-colonies are called u. The algorithm will reconstruct the pheromone matrix of u which is immersed in local optimum according to the Pearson correlation coefficient. The selection of sub-colonies u is shown as follows:

$$u = \begin{cases} U, & if\ H(U) > h_0 \ \wedge \ D(H(U)){>}w_0 \ \wedge \ 1 \le U \le M \\ Null, & otherwise \end{cases} \tag{13}$$

Where U is the number of sub-colony, $H(U)$ is the similarity between the optimal path of colony U and the minimum spanning tree, h_0 is a threshold, $D(H(U))$ is the number of iterations in which the value of $H(U)$ remains unchanged and w_0 is a threshold and M is the total number of sub-colonies.

If u is not null in the current iteration, the algorithm is judged to fall into the local optimum. At this time, the dynamic interaction strategy needs to be implemented in the current iteration, and the details of the interaction mechanism will be described in section 4.2.2. In other words, the frequency of colony communication is determined by the frequency with which u is generated, so that neither too frequent communication nor too early disruption of the search experience within the colony. To sum up, this method helps the sub-colony to adaptively adjust the communication frequency so that the algorithm breaks the stagnation state.

4.2.2 Dynamic interaction strategy

When the algorithm falls into local optimization, a simple method is to reset the pheromone concentration of each edge to the initial pheromone value. For example, MMAS judges whether the algorithm is close to the stagnation state by calculating the pheromone size of each edge or failing to get a better path in a specified number of iterations, and then the pheromone is initialized. However, this method not only fails to retain the experience accumulated by the algorithm in the past, but also slows down the convergence speed of the algorithm for large-scale problems.

Collaborative filtering uses the preferences of some groups with similar interests and common experiences to recommend information that users are interested in. It makes predictions and recommendations based on the ratings or behavior of other users in the system. In this paper, we use the collaborative filtering algorithm to recommend the values in the pheromone matrix of the target sub-colonies, which can not only retain part of the target sub-colony's own experience, but also learn the differential search experiences recommended by other similar sub-colonies. Thus, the search area can be expanded directly and effectively, and the accuracy of the solution can be improved. More specifically, we can use the pheromone concentration of the corresponding paths of similar sub-colonies to recommend the pheromone concentration of the corresponding paths of the target sub-colonies. If the similarity value is higher, the more the target sub-colonies retain their own search experience and the less differential search experience is recommended by other sub-colonies. If the similarity value is lower, the less the target sub-colonies retain their own search experience and the more differential search experience is recommended by other sub-colonies. In this way, the pheromone matrix of the target sub-colonies is reconstructed, so that they can effectively jump out of the local optimum. The target sub-colonies are the sub-colonies that fall into the local optimum. To implement collaborative filtering, three steps are needed. The first step is to collect user preferences. The second step is to find similar users or items. The third step is to predict and recommend. The above steps are corresponding to the steps in this paper are as follows: the first step is to find the iterative optimal solution of the sub-colonies so far; the second step is to find the similar sub-colonies of the target sub-colonies; the third step is to recommend the pheromone concentration on the corresponding path of the target sub-colonies according to the information of the similar sub-colonies.

In a collaborative filtering algorithm, the Pearson correlation coefficient is an important method to measure the degree of correlation between two variables. The higher the correlation coefficient between two variables, the higher the accuracy of prediction from one variable to the other variable, which means that the two variables will have more covariant parts, so the more chances of one variable can be known about the changes of the other variable.

The range of the Pearson correlation coefficient is (-1, 1). It becomes a complete positive correlation when the correlation coefficient is 1. When the correlation coefficient is -1, it becomes a completely negative correlation. The greater the absolute value of the correlation coefficient, the stronger the correlation. The closer the correlation coefficient is to 0, the weaker the correlation.

However, the rating matrix is the basis of the collaborative filtering algorithm, which is mainly used to collect and store information about user preferences. We can obtain the Pearson correlation coefficient according to the rating matrix. In this paper, the Pearson correlation coefficient is calculated according to the pheromone concentration of the common path, in which the common path comes from similar parts of the optimal solution of the sub-colonies. The rating matrix can be represented by matrix *R(m*n)*, as shown in Table 2:

Where, P_i represents the sub-colony *i*, R_j represents the path *j*, and Po_{ij} represents the pheromone concentration of the sub-colony *i* on the path *j*.

In this method, the Pearson correlation coefficient is calculated by (9). The selection method of the colony falling into the local optimum is given by (13). Then the users are sorted according to the degree of similarity, and *k* users with the highest similarity are selected as the k-nearest neighbor set. However, the probability of the target sub-colonies exploring new paths and exploiting the original paths is dynamically adjusted by the k-Nearest Neighbor set. The larger the value of *k*, the lower the probability that the target sub-colonies retain the original search path, and the higher the probability of the differential search path recommended by other sub-colonies. The smaller the value of *k*, the higher the probability that the target sub-colonies retain the original search path, and the lower the probability of the differential search path recommended by other sub-colonies. More specifically, sub-colonies are explored and exploited at the same time. Therefore, the algorithm can achieve the purpose of effectively balancing exploration and exploitation.

Table 2 The rating matrix of the colony

	R_1	...	R_j	...	R_n
P_1	Po_{11}	...	Po_{1j}	...	Po_{1n}
...	...	...	...	...	...
P_i	Po_{i1}	...	Po_{ij}	...	Po_{in}
...	...	...	...	...	...
P_m	Po_{m1}	...	Po_{mj}	...	Po_{mn}

When the Pearson correlation coefficient is 1 or -1, the relationship between colonies is a completely positive correlation or negative correlation respectively. Therefore, if the Pearson correlation coefficient is other values, the predictive pheromone has some error, that is, the problem of credibility. We take the following measures to solve the problem. First of all, the probability of sub-colony *u* communicating with each other is s_g by the Pearson correlation coefficient. s_g is calculated by the (14). Meanwhile, the pheromone matrix predicted by the Pearson correlation coefficient is inaccurate. So the algorithm will conduct pheromone communication between sub-colony *u* and sub-colony *h* with the probability of (1-s_g). *h* is the sub-colony with the highest information entropy. More specifically, the higher the similarity, the higher the probability that the target sub-colonies can reconstruct the pheromone matrix with the Pearson correlation coefficient, and the lower the similarity, the higher the probability that the target sub-colonies exchange information with the sub-colonies with the highest information entropy. Because the higher the similarity, the more accurate the recommended value will be. In this way, the effectiveness of the interaction is guaranteed, and the target sub-colonies are easier to jump out of the local optimum.

Pheromone updates on the relevant paths of sub-colony *u* are shown as follows:

$$s_g = \max(|sim(u,1)|, |sim(u,2)|, \cdots, |sim(u,n)|) \quad (14)$$

Where, $sim(u, n)$ represents the Pearson correlation coefficient between the sub-colony *u* and the *n-th* sub-colony.

$$\underset{j \in N_R}{p_{u,j}} = \begin{cases} \bar{r}_u + \frac{\sum\limits_{v \in N(u)} sim(u,v) \times (r_{v,j} - \bar{r}_v)}{\sum\limits_{v \in N(u)} sim(u,v)} & , s_0 \le s_g \\ p_{h,j}, & else \end{cases} \quad (15)$$

Where, N_R represents the set of paths between cities, $p_{u,j}$ represents the pheromone concentration on the path *j* of colony *u*, $\bar{r}_u$ represents the mean value of the pheromone concentration of the unrepeated path in the pheromone matrix of sub-colony *u*, $\bar{r}_v$ represents the mean value of the pheromone concentration of the unrepeated path in the pheromone matrix of sub-colony *v*, *N(u)* represents the k-Nearest Neighbor set of target colony *u*, and s_0 is a random number between 0 and 1.

In this paper, the adaptive interaction frequency strategy determines the frequency of communication, and the dynamic interaction strategy determines which colonies communicate and how to communicate. According to the search progress of the sub-colonies themselves, the two strategies adaptively organize the interaction tasks of the sub-colony, which guarantees the effectiveness of the interaction, strengthens the adaptability of the algorithm, and also helps to improve the accuracy of the solution.

4.3 Algorithm framework

PPACO Algorithm for TSP

```
Initialize the pheromone and parameters
Calculate the distance between cities
While termination condition is not satisfied do
    If n ≤ $n_0$
        Update spanning-tree pheromone for DGMACO with (11), (12)
    end-if
    Construct ant solution for ACS and DGMACO with (1), (2)
    Construct ant solution for MMAS with (2)
    Update pheromone for ACS and DGMACO with (3), (4)
    Update pheromone for MMAS with (5), (6), (7)
    Calculate the $u$ with (13)    /* $u$ is the sub-colony that is locally optimal*/
    If $u$ is not null
        Calculate the similarity with (9)
        Update the pheromone matrix of $u$ with (14), (15)
    end-if
    NC = NC + 1
END while
Comparing the optimal solutions of DGMACO, ACS, and MMAS
Output the minimum solution
```

The above is the pseudo-code of the algorithm in this paper. Figure 1 is the basic framework of the algorithm in this paper. The multi-colony ant colony algorithm proposed in this paper is composed of three types of single colony algorithms: ACS, MMAS, and DGMACO. ACS is responsible for accelerating convergence, MMAS is responsible for improving diversity and DGMACO is responsible for balancing convergence and diversity. In the beginning, sub-colonies carry out path optimization and pheromone update operations according to their respective mechanisms. When the adaptive communication conditions are met, the relevant sub-colonies are selected to communicate with each other, which helps the colony jump out of the local optimum and obtain high-precision solutions.

4.4 The time complexity of the algorithm

From the analysis of the above algorithm pseudo-code, it can be concluded that the number of executions of the PPACO algorithm is $n * N * k * m$. Where n is the number of sub-colonies, it is a constant, N is the maximum number of iterations, k is the total number of ants in each sub-colony, and m is the number of cities. So the maximum time complexity of PPACO is $o(N * k * m)$. However, the maximum time complexity of ACS is $o(N * k * m)$ and the maximum time complexity of MMAS is $o(N*k*m)$. It can be seen that compared with ACS and MMAS, the PPACO algorithm does not change the maximum time complexity.

5 Experiment and simulation

The experiments were simulated in MATLAB R2016a on an Intel Core-i5 PC. To effectively verify the performance of PPACO, the experiments in this paper selected several cities from the standard TSPLIB database for systematic analysis. Meanwhile, it also made a comprehensive comparison with the classical ACS and MMAS algorithms to analyze the advantages and disadvantages of the proposed algorithm. Finally, PPACO is compared with other improved ant colony algorithms and other intelligent algorithms to further test the performance of the proposed algorithm.

5.1 Parameter setting of PPACO algorithm

The scientific method of designing experiments should not only minimize the number of experiments as much as possible, but also make use of the obtained experimental data on the basis of a small number of experiments to analyze the correct conclusions and get better results. The orthogonal experiment can select a few test schemes with strong representativeness evenly, and introduce a better scheme among the few test results. Therefore, in order to make the PPACO have better performance, the orthogonal test of three levels and four factors is used to determine the parameters in ACS, the orthogonal test of three levels and three factors is used to determine the parameters in MMAS, and the orthogonal test of two levels and seven factors is used to determine the parameters in DGMACO, so as to make the algorithm obtain a better combination of parameters. As other orthogonal experiments, the values in this paper are obtained through the preliminary optimization phase. We ran every program 15 times and further calculated the average of TSP eil76.

Based on the experimental results in Tables 3, 4, 5, 6, 7, 8, 9, 10 and 11. We know that in ACS, the best combination

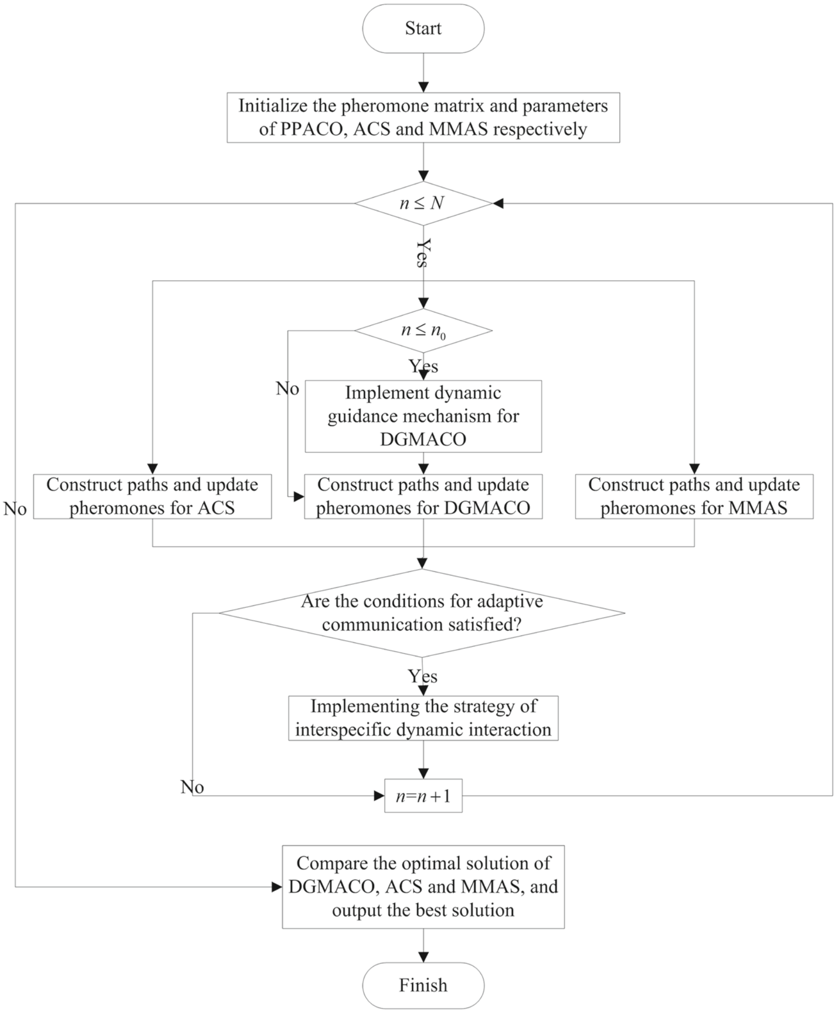

Start
Initialize the pheromone matrix and parameters of PPACO, ACS and MMAS respectively
$n \le N$
Yes
No
$n \le n_0$
Yes
No
Implement dynamic guidance mechanism for DGMACO
Construct paths and update pheromones for ACS
Construct paths and update pheromones for DGMACO
Construct paths and update pheromones for MMAS
Are the conditions for adaptive communication satisfied?
Yes
No
Implementing the strategy of interspecific dynamic interaction
$n=n+1$
Compare the optimal solution of DGMACO, ACS and MMAS, and output the best solution
Finish

Fig. 1 Flowchart of the PPACO

of parameters is that α takes the value of 1, β takes the value of 4, ρ takes the value of 0.3, ξ takes the value of 0.1, in MMAS, the best combination of parameters is that α takes the value of 1, β takes the value of 3, ρ takes the value of 0.2, and in DGMACO, the best combination of parameters is that α takes the value of 1, β takes the value of 4, ρ takes the value of 0.4, ξ takes the value of 0.2, ρ_b takes the value of 1, ρ_s takes the value of 1, n_0 takes the value of $N/_3$.

Table 3 Experimental Factors and Levels of ACS

	α	β	ρ	ξ
Level 1	1	2	0.2	0.1
Level 2	2	3	0.3	0.2
Level 3	3	4	0.4	0.3

$^*\rho$ is the globally updated evaporation factor in the ACS, and ξ is the locally updated evaporation factor in the ACS.

5.2 Statistical test of the algorithm

Because ant colony optimization is a probability algorithm, we can only do a limited number of experiments. However,

Table 4 Orthogonal test scheme and test results of ACS

no.	α	β	ρ	ξ	*results*
1	1	2	0.2	0.1	548.1
2	1	3	0.3	0.2	544.0
3	1	4	0.4	0.3	543.1
4	2	2	0.3	0.3	551.3
5	2	3	0.4	0.1	543.2
6	2	4	0.2	0.2	543.6
7	3	2	0.4	0.2	550.0
8	3	3	0.2	0.3	549.3
9	3	4	0.3	0.1	539.7

**no.* is the number of tests, and *results* are the average of 15 test data in each group.

Table 5 Analysis of test results of ACS

K	α	β	ρ	ξ
K_1	1635.20	1649.40	1641.00	1631.00
K_2	1638.10	1636.50	1635.00	1637.60
K_3	1639.00	1626.40	1636.30	1643.70
k_1	545.07	549.80	547.00	543.67
k_2	546.03	545.50	545.00	545.87
k_3	546.33	542.13	545.43	547.90
max	546.33	549.80	547.00	547.90
min	545.07	542.13	545.00	543.67
range	1.26	7.67	2	4.23
scheme	*Level* 1	*Level* 3	*Level* 2	*Level* 1

*$K_i(i = 1,\ 2,\ 3)$ are the sum of results. $k_i(i = 1,\ 2,\ 3)$ are the means of every level. *max* and *min* are the results of maximal and minimal path length respectively. *range* is the difference by the *max* minus the *min*, which will be applied to determine which one factor is important. *scheme* is the project of every factor by orthogonal test to obtain the best result.

Table 6 Experimental factors and levels of MMAS

	α	β	ρ
Level 1	1	1	0.1
Level 2	2	2	0.2
Level 3	3	3	0.3

*ρ is the evaporation factor in the MMAS.

Table 7 Orthogonal test scheme and test results of MMAS

no.	α	β	ρ	*results*
1	1	1	0.1	557.9
2	1	2	0.2	554.4
3	1	3	0.3	545.8
4	2	1	0.2	562.5
5	2	2	0.3	557.5
6	2	3	0.1	551.1
7	3	1	0.3	564.4
8	3	2	0.1	559.2
9	3	3	0.2	550.2

Table 8 Analysis of test results of MMAS

K	α	β	ρ
K_1	1658.10	1684.80	1668.20
K_2	1671.10	1671.10	1667.10
K_3	1673.80	1647.10	1667.70
k_1	552.70	561.60	556.07
k_2	557.03	557.03	555.70
k_3	557.93	549.03	555.90
max	557.93	561.60	556.07
min	552.70	549.03	555.70
range	5.23	12.57	0.37
scheme	*Level* 1	*Level* 3	*Level* 2

Table 9 Experimental factors and levels of DGMACO

	α	β	ρ	ξ	ρ_b	ρ_s	n_0
Level 1	1	3	0.3	0.1	1	1	$N/_2$
Level 2	2	4	0.4	0.2	2	2	$N/_3$

*ρ is the globally updated evaporation factor in the DGMACO, and ξ is the locally updated evaporation factor in the DGMACO. ρ_b represents the maximum spanning tree adjustment factor, ρ_s represents the minimum spanning tree adjustment factor, n_0 is the threshold for the number of iterations, N is the maximum number of iterations of the algorithm.

Table 10 Orthogonal test scheme and test results of DGMACO

no.	α	β	ρ	ξ	ρ_b	ρ_s	n_0	*results*
1	1	3	0.3	0.1	1	1	$N/_2$	543.9
2	1	3	0.3	0.2	2	2	$N/_3$	545.2
3	1	4	0.4	0.1	1	2	$N/_3$	542.6
4	1	4	0.4	0.2	2	1	$N/_2$	541.0
5	2	3	0.4	0.1	2	1	$N/_3$	547.2
6	2	3	0.4	0.2	1	2	$N/_2$	547.5
7	2	4	0.3	0.1	2	2	$N/_2$	547.2
8	2	4	0.3	0.2	1	1	$N/_3$	543.1

Table 11 Analysis of test results of DGMACO

K	α	β	ρ	ξ	ρ_b	ρ_s	n_0
K_1	2172.70	2183.80	2179.40	2180.90	2177.10	2175.20	2179.60
K_2	2185.00	2173.90	2178.30	2176.80	2180.60	2182.50	2178.10
k_1	543.18	545.95	544.85	545.23	544.28	543.80	544.90
k_2	546.25	543.48	544.58	544.20	545.15	545.63	544.53
max	546.25	545.95	544.85	545.23	545.15	545.63	544.90
min	543.18	543.48	544.58	544.20	544.28	543.80	544.53
range	3.07	2.47	0.27	1.03	0.87	1.83	0.37
scheme	*Level* 1	*Level* 2	*Level* 2	*Level* 2	*Level* 1	*Level* 1	*Level* 2

when we analyze the performance difference between algorithms through the experimental results, we cannot judge whether the difference is purely chance variation or caused by the improvement work we have done. So we need to carry out a significance test to check whether the algorithm proposed in this paper is significantly different from the traditional ant colony optimization and other improved ant colony optimization. Because the Friedman test does not require the assumption of normality and homogeneity of variance, the Friedman test is used to test the significance in this paper. EDHACO is one of the other published improved ant colony algorithms [46].

Friedman test only focuses on whether there are significant differences between the levels of each column, and it is not interested in the area groups of each row at all. So to make the experiment more reasonable, we selected the large-scale TSP instance lin318, the medium-scale TSP instance kroa200, the small-scale TSP instance eil76 as the experimental objects. Besides, we selected 10 experimental data from each scale instance to carry out the Friedman test in spss25 software.

First, we give the original hypothesis, H_0 : there is no significant difference in the performance of the four algorithms. Then, we input the data into spss25 software and get the final result chart. The significance level in Table 12 is $p = 0 < 0.05$, so the decision is to reject the null hypothesis. More specifically, the performance of the four algorithms is significantly different. It can be seen from Fig. 2 that the mean rank of PPACO is 1.00, the mean rank of MMAS is 3.95, the mean rank of ACS is 2.08, and the mean rank of EDHACO is 2.97. Pairwise comparisons are needed because of the difference in response rates at different frequencies. The results of the pairwise comparison are shown in Table 13. As can be seen from Table 13, the Adj. Sig of PPACO and ACS was 0.007 < 0.05. The Adj. Sig of PPACO and MMAS was 0 < 0.05. The Adj. Sig of PPACO and EDHACO was 0 < 0.05. In conclusion, PPACO is different from ACS, MMAS, and EDHACO. In other words, the performance comparison between PPACO and other algorithms has statistical significance in the following experiments.

Table 12 Hypothesis Test Summary

Null Hypothesis	Test	Sig.	Decision
The distributions of PPACO, MMAS, ACS and EDHACO are the same.	Related-Samples Friedman's Two-Way Analysis of Variance by Ranks	0	Reject the null hypothesis.

*The significance level is 0.05.

5.3 Performance test of PPACO

5.3.1 Analysis of experimental results of PPACO

To compare the performance of ACS, MMAS, and PPACO in multiple directions, 13 TSP instances of different scales were selected in this paper. Each experiment was performed by 2,000 iterations. The experimental analysis is carried out from the following aspects: the optimal solution (Best), the average solution (Mean), standard deviation (dev), the minimum error rate (Error rate), the convergence iteration (Convergence). The experimental data is in Table 14. Equation (16) is the specific solution formula of standard deviation.

$$dev = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_i - L_r)^2} \tag{16}$$

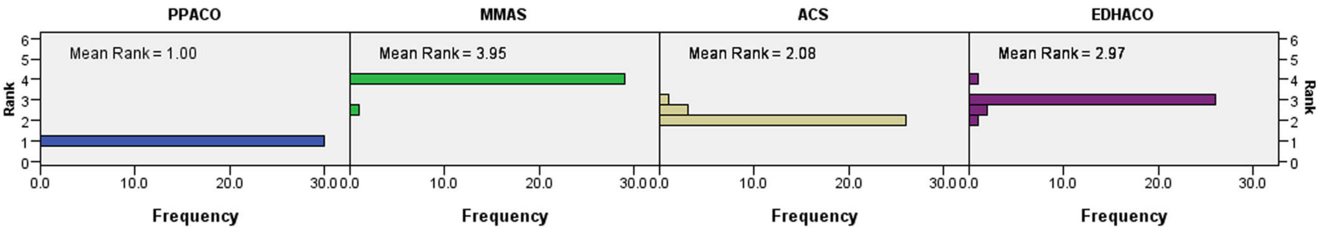


Fig. 2 Related-Samples Friendman's Two-Way Analysis of Variance by Ranks

Where N is the number of experiments per TSP instance (N= 15), L_i is the optimal solution for each experiment, L_r is the average solution for N experiments.

The results listed in Table 14 can be analyzed more specifically. On small scale TSP instances with less than 100 cities, the PPACO can find the optimal solution faster than ACS and MMAS. Because of the minimum spanning tree strategy in the dynamic guidance mechanism, the PPACO can get a faster convergence rate. On the medium scale TSP instances, such as kroB150 ch150, kroA200, PPACO can obtain the standard optimal solution. Because of the pheromone refactoring mechanism based on Pearson correlation coefficient, the PPACO jumps out of the local optimum. For tsp225, pr226, and a280, although PPACO does not find the standard optimal solution, the accuracy of the optimal solution is higher than that of ACS and MMAS. It demonstrates that the quality of the solution can be improved by the pheromone refactoring mechanism based on Pearson correlation coefficient. In large scale TSP instances with more than 300 cities, the PPACO is difficult to find the standard optimal solution, but its convergence rate and solution accuracy are better than those of ACS and MMAS. Moreover, the error rates of PPACO can be kept within 1%. These results prove that the dynamic guidance mechanism and the pheromone refactoring mechanism based on Pearson correlation coefficient in the proposed algorithm can ensure the solution accuracy while accelerating the convergence rate.

Besides, the error rates of different scales of instances are depicted in Fig. 3. We can intuitively see that the error rate of PPACO is much lower than that of ACS and MMAS in TSP instances of different scales. This further proves that PPACO can obtain more accurate solutions.

Standard deviation can reflect the dispersion degree of a data set, which represents the stability of the algorithm in this paper (16). In the 13 TSP instances in Fig. 4, the column height of PPACO is lower than that of ACS and MMAS. It can be seen from the dev column data in Table 14 that the standard deviations of PPACO are lower than those of ACS and MMAS. It indicates that the stability of PPACO is better than that of ACS and MMAS in TSP instances of different scales.

5.3.2 Convergence comparison and optimal path graph of PPACO

The convergence curves of different scales of instances are also depicted in Fig. 5, where we take the instance eil51, kroA100, kroB150, kroA200, pr264, and lin318 as examples to illustrate the convergence ability of our proposed algorithm. Figure 5 demonstrates that in small scale and medium scale instances, PPACO not only converges faster than ACS and MMAS, but also can

Table 13 Multiple comparison

Sample1-Sample2	Test Statistic	std. Error	Std. Test Statistic	Sig.	Adj.Sig
PPACO-ACS	-1.083	0.333	-3.250	0.001	0.007
PPACO-EDHACO	-1.967	0.333	-5.900	0	0
PPACO-MMAS	-2.950	0.333	-8.850	0	0
ACS-EDHACO	-0.883	0.333	-2.650	0.008	0.048
ACS-MMAS	1.867	0.333	5.600	0	0
EDHACO-MMAS	0.983	0.333	2.950	0.003	0.019

Table 14 Performance comparison of PPACO, ACS, MMAS in different TSP instances

Instances	Opt	ACO	Best	Mean	dev	Error rate(%)	Convergence
Eil51	426	ACS	426	428	1.08	0	1092
		MMAS	428	431	2.94	0.46	1582
		PPACO	426	427	0.48	0	277
Eil76	538	ACS	538	543	4.58	0	1528
		MMAS	543	550	3.46	0.92	1765
		PPACO	538	539	2.08	0	177
kroA100	21282	ACS	21282	21316	43.53	0	1538
		MMAS	21349	21568	151.51	0.31	1990
		PPACO	21282	21291	9.61	0	249
kroB150	26130	ACS	26161	26341	90.22	0.10	923
		MMAS	26878	27197	236.03	2.86	1953
		PPACO	26130	26232	62.82	0	596
ch150	6528	ACS	6543	6577	21.86	0.22	1125
		MMAS	6655	6744	66.59	1.94	1481
		PPACO	6528	6556	14.95	0	329
kroA200	29368	ACS	29495	29626	138.85	0.43	1297
		MMAS	29803	30359	353.64	1.48	1424
		PPACO	29368	29480	77.50	0	1075
tsp225	3916	ACS	3944	3995	23.86	0.72	1880
		MMAS	4104	4192	105.80	4.80	1986
		PPACO	3935	3957	15.07	0.40	219
pr226	80369	ACS	80618	80881	337.15	0.30	1332
		MMAS	81852	82531	409.15	1.84	1148
		PPACO	80381	80450	35.43	0.01	1257
pr264	49135	ACS	49645	50335	881.84	1.03	1773
		MMAS	51430	52662	621.24	4.67	1741
		PPACO	49135	49183	24.17	0	670
a280	2579	ACS	2605	2636	22.57	1.00	1885
		MMAS	2722	2784	126.87	5.54	1987
		PPACO	2582	2601	15.12	0.10	1684
lin318	42029	ACS	43155	43650	297.14	2.67	1979
		MMAS	44794	45285	534.43	6.57	1881
		PPACO	42214	42832	297.02	0.40	963
fl417	11861	ACS	12193	12330	254.28	2.80	1989
		MMAS	12664	13116	254.267	6.77	1890
		PPACO	11957	11999	25.75	0.80	307
pr439	107217	ACS	109037	110650	4540.90	1.70	1906
		MMAS	117104	122830	6935.30	9.22	1808
		PPACO	108263	108983	321.78	0.97	1866

accurately and quickly find the standard optimal solutions of eil51, kroA100, kroB150, kroA200. In large scale instances, the convergence speed of the PPACO is still better than that of ACS and MMAS. Although it does not find the standard optimal solution, the accuracy of the optimal solution obtained is much higher than that of ACS and MMAS. These results prove that the minimum spanning tree strategy in the dynamic guidance mechanism can accelerate the convergence of the algorithm directionality. Besides, the PPACO adopts the pheromone refactoring mechanism, which makes the algorithm easy to jump out of the local optimum and find the value closer to the standard optimal solution.

To verify the authenticity of the optimal solution obtained by the algorithm, Fig. 6 illustrates the tours of optimal solutions found by our algorithm in several TSP instances.

5.4 Performance Analysis of Improved Mechanism

5.4.1 Dynamic Guidance Mechanism Experiment

To verify the performance advantages of a dynamic guidance mechanism to the algorithm, a small-scale instance kro100, a medium-scale instance kroA200 and a large-scale instance lin318 were selected for convergence analysis in this paper. Each experiment was performed 15 times and the optimal solution was selected to compare the convergence. L-DGMACO is a PPACO that removes the dynamic guidance mechanism, that is, PPACO removes the single colony DGMACO. Figure 7 shows that the convergence rate of PPACO is faster than that of the L-DGMACO, and the convergence accuracy is also improved. Because of the high similarity between the minimum spanning tree and the standard optimal path, the minimum spanning tree strategy in the dynamic guidance mechanism adaptively strengthens the pheromone concentration on the relevant paths, thus accelerating the directional convergence of the colony. Besides, due to the dominance of distance heuristic information in the initial iteration, the maximum spanning tree strategy dynamically adjusts the pheromone concentration of the relevant paths, which increases the diversity of the algorithm and leads to improved convergence accuracy. Therefore, the dynamic guidance strategy plays a very good role in balancing the diversity and convergence of the algorithm.

5.4.2 Pheromone refactoring mechanism experiment

The improved algorithm can find the optimal solution on the small test set and is not easily immersed in local optimization. To prove the effectiveness of the pheromone refactoring mechanism based on Pearson correlation coefficient, two medium-scale instances kroA200, pr264, and a large-scale instance lin318 were selected in this paper as the precision analysis of solution. Each experiment was performed 15 times and the optimal solution was selected to compare the convergence. L-PPACO is a PPACO that removes the pheromone refactoring mechanism based on Pearson correlation coefficient. Figure 8 demonstrates the convergence speed of the PPACO is faster than that of the L-PPACO. More importantly, the L-PPACO produces a value of the optimal solution inferior to that of the PPACO for the accuracy of the solution, which indicates that the pheromone refactoring mechanism based on Pearson correlation coefficient can make the PPACO easily jump out of local optimum, and greatly improve the quality of the solution. On the other hand, Fig. 9 shows that the error rate of the PPACO is lower than that of the L-PPACO in TSP instances of different scales, which also proves that the PPACO can improve the accuracy of the solution. Because the pheromone reconstruction mechanism based on Pearson correlation coefficient can not only make the sub-colony have a certain probability to retain the original search paths, but also have a certain probability to learn the different search paths recommended by other sub-colonies. This will be beneficial to the exploitation and exploration of the sub-colony at the same time, so as to effectively balance the exploitation and exploration.

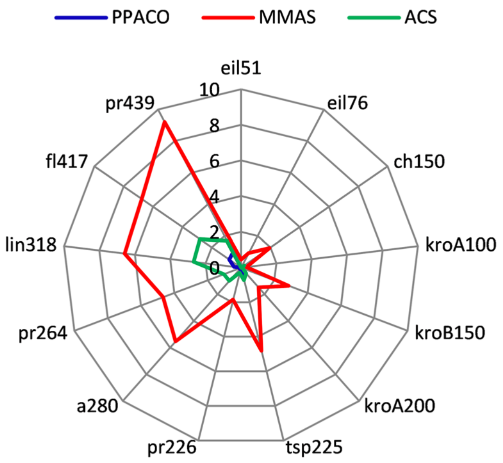


Fig. 3 Comparison chart of ACS, MMAS, and PPACO error rate

5.4.3 Comprehensive performance analysis

To further validate the effectiveness of the dynamic guidance mechanism and pheromone refactoring mechanism on the algorithm, we selected kroA200 and lin318 as experimental objects. The two TSP instances were tested 15 times. The experimental analysis is carried out from the following aspects: the optimal solution (Best), the average solution

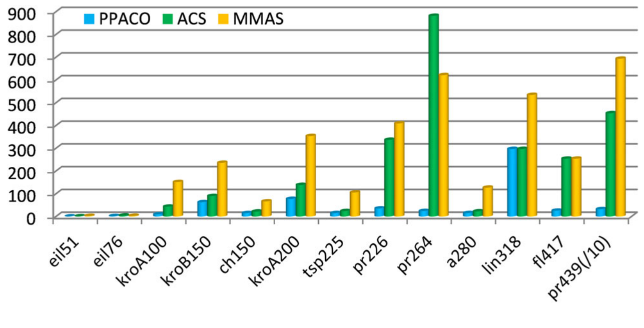
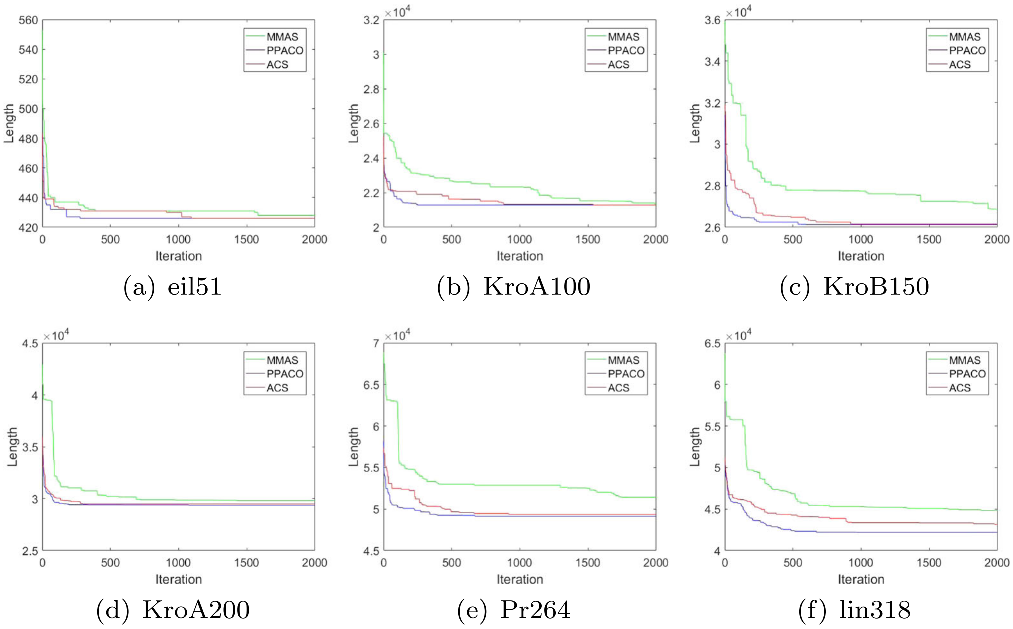


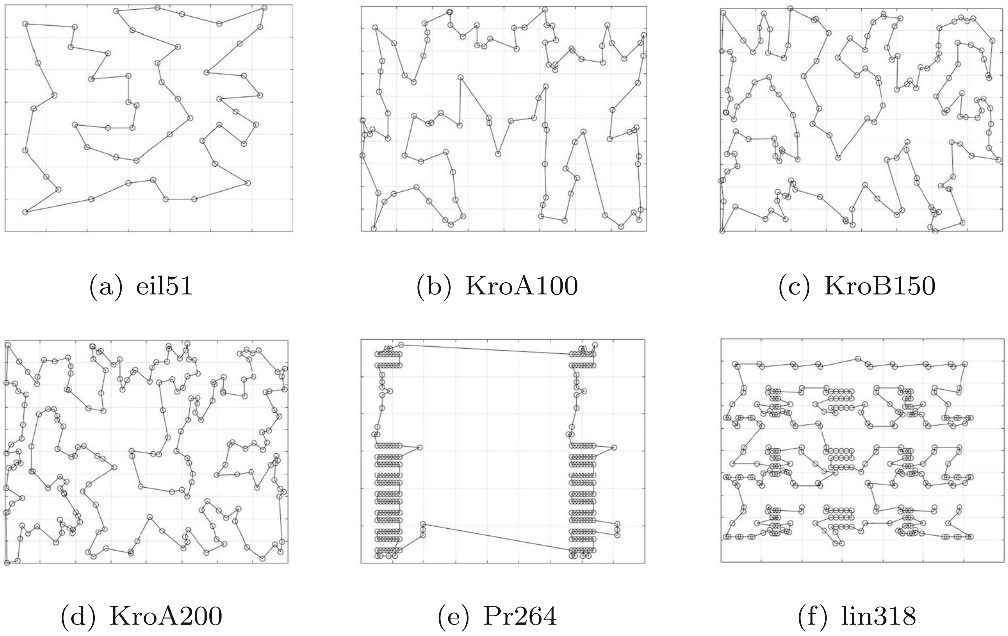
Fig. 4 Comparison of the stability of different algorithms



(a) eil51 (b) KroA100 (c) KroB150

(d) KroA200 (e) Pr264 (f) lin318

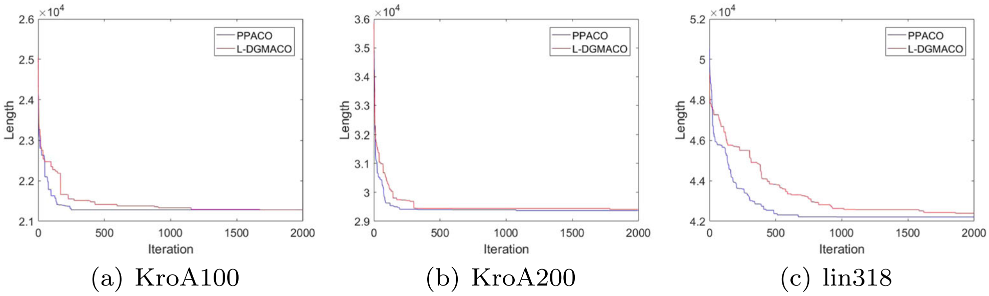
Fig. 5 Comparison of convergence of ACS, MMAS, and PPACO



(a) eil51 (b) KroA100 (c) KroB150

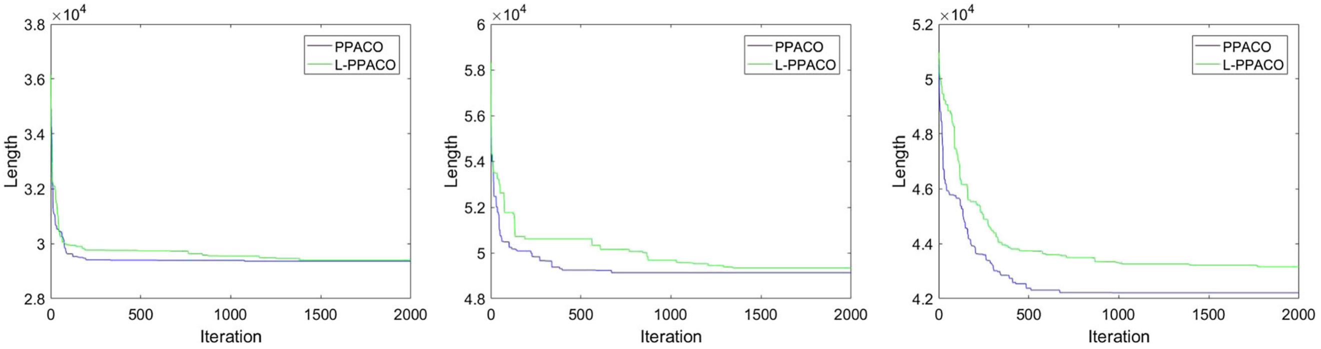
(d) KroA200 (e) Pr264 (f) lin318

Fig. 6 Best tours for each TSP instance found by PPACO



(a) KroA100 (b) KroA200 (c) lin318

Fig. 7 Comparison convergence of PPACO and L-DGMACO



(a) KroA200 (b) pr264 (c) lin318

Fig. 8 Comparison of convergence of PPACO and L-PPACO

(Mean), the minimum error rate (Error rate), the convergence iteration (Convergence). The experimental data is in Table 15. L-DGMACO is an improved algorithm with a pheromone refactoring mechanism but lacks a dynamic guidance mechanism. L-PPACO is an improved algorithm with a dynamic guidance mechanism but lacks a pheromone refactoring mechanism.

Firstly, we analyze whether each mechanism effectively improves the algorithm. It can be seen in Table 15, all three improved algorithms can find the standard optimal solution in kroA100. Although the ACS also finds the standard optimal solution, the average value of the three improved ant colony algorithms is better than that of the ACS and MMAS. Only PPACO can find the standard optimal solution in pr264, but the optimal and average solutions of the three improved algorithms are still better than ACS and MMAS. Besides, it can be seen in Fig. 10 that the three improved algorithms converge faster than ACS and MMAS. Therefore, it can be understood that the accuracy and convergence rate of the three improved algorithms are better than that of ACS and MMAS. In other words, the dynamic guidance mechanism and pheromone refactoring mechanism based on Pearson correlation coefficient play an effective role in the performance of the algorithm.

Then, the main effects of each mechanism on the algorithm are analyzed. It can be seen from Table 15 that the quality of the optimal solution and average solution of the L-PPACO is worse than that of PPACO and L-DGMACO. In addition, in pr264 instance, PPACO can find the standard optimal solution, and L-DGMACO finds the optimal solution is very close to the standard optimal solution. It indicates that the pheromone refactoring mechanism based on Pearson correlation coefficient is helpful for the algorithm to improve the accuracy of the solution and jump out of the local optimal. According to Fig. 10, it can be seen that the convergence curves of L-DGMACO are slower than that of the other two improved algorithms in two TSP instances. It can be seen from Fig. 11 that in the early iteration of pr264 instance, the curve of L-DGMACO is outside the curve of the other two improved algorithms. It proves that the dynamic guidance mechanism can effectively accelerate the convergence of the algorithm.

Finally, Figs. 10 and 11 and Table 15 show that the convergence of PPACO is the fastest, and its solution is also the most accurate. Therefore, the dynamic guidance mechanism and pheromone refactoring mechanism based on Pearson correlation coefficient jointly improve the performance of the algorithm, so that the solution accuracy and convergence speed of PPACO is improved.

5.5 Compared with other improved algorithms in TSP

To further illustrating the performance of our proposed algorithm, we compare it with several improved algorithms. Table 16 shows the data of PPACO and other improved algorithms. EDHACO, PACO-3Opt and PCCACO are improved multi-colony ant colony algorithms. PSO-ACO-3Opt, AS-SA-Opt, and HACO are hybrid algorithms based

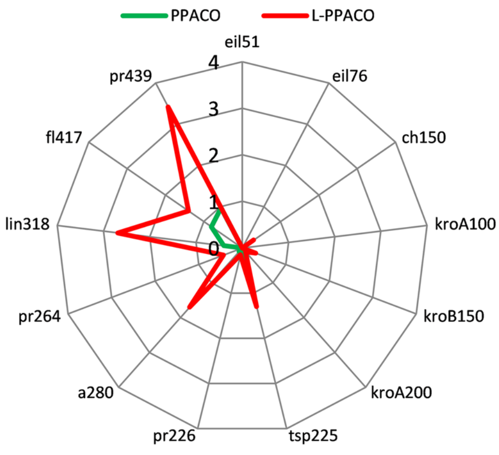


Fig. 9 Comparison chart of L-PPACO and PPACO error rate

Table 15 Performance analysis of algorithms with different mechanisms

Instances	ACO	Best	Mean	Error rate(%)	Convergence
kroA100	PPACO	21282	21291	0	249
	L-DGMACO	21282	21313	0	787
	L-PPACO	21282	21320	0	349
	ACS	21282	21336	0	1538
	MMAS	21349	21609	0.31	1990
pr264	PPACO	49135	49183	0	670
	L-DGMACO	49212	49351	0.15	1601
	L-PPACO	49295	49515	0.32	780
	ACS	49645	50335	1.03	1773
	MMAS	51430	52662	4.67	1741

on ant colony algorithm. GA, FOA, PSO, and ABC are other intelligent algorithms. The data refers to its relevant literature, and the number in parentheses after the algorithm name indicates the location of the reference. *Best* is the optimal path length of the relevant algorithm, *Mean* is the average path length of the algorithm, and $PD\_Best(\%)$ is the minimum error rate of the algorithm.

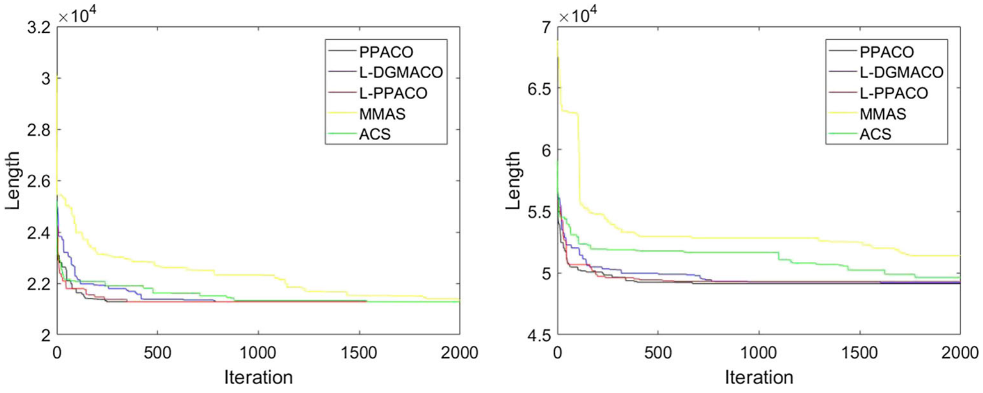
As shown in Table 16, for small scale and medium scale TSP instances with less than 200 cities, PPACO can all find the standard optimal solution, while not all other improved algorithms can find the standard optimal solution. The average solution of the PPACO is better than most of the other algorithms. For medium scale and large scale TSP instances with more than 200 cities, the accuracy of solutions found by the PPACO is better than that of other multi-colony ant colony algorithms, hybrid algorithms based on ant colony algorithm and other intelligent algorithms. Especially for large-scale problems, the optimal solution of PPACO is closer to the standard optimal solution than other improved algorithms, and its error rate can be kept within 1%.

Through a series of experimental analysis, we can see that the PPACO has certain advantages over other improved algorithms. PPACO not only speeds up the convergence speed, but also improves the accuracy of the solution.

6 Conclusion

In this paper, an ant colony optimization based on a dynamic guidance mechanism is proposed as a single colony, which is combined with ACS and MMAS to form a multi-colony ant colony algorithm. In the early stage of the algorithm, the maximum spanning tree strategy is introduced to increase the pheromone on the path of the maximum spanning tree adaptively, which can improve the search ability of the algorithm. Meanwhile, the minimum spanning tree strategy

Fig. 10 Comparison of convergence of algorithms with different mechanisms



(a) KroA100 (b) pr264

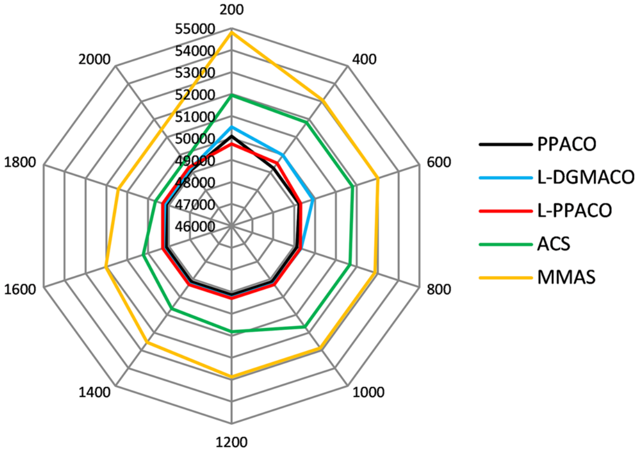


Fig. 11 Iterative breakpoint graph of pr264 with different algorithms

is introduced to increase the pheromone on the path of the minimum spanning tree adaptively, which can accelerate the convergence rate of the algorithm.These two strategies complement each other and can effectively balance diversity and convergence.

In addition, pheromone refactoring mechanisms are defined to ensure effective communication between sub-colonies. First, an adaptive interaction frequency strategy is used to adjust the frequency of communication among sub-colonies. Then, the dynamic interaction strategy can reconstruct the pheromone matrix of the sub-colonies. The sub-colonies can choose the original path or the differential path recommended by other sub-colonies, which can effectively balance exploration and exploitation. Therefore, under the joint action of the two strategies, the algorithm

Table 16 Comparison with other improved algorithms in TSP

ACO		Eil51	Eil76	KroA100	KroB150	Ch150	KroA200	Lin318	Pr439
	Opt	426	538	21282	26130	6528	29368	42029	107217
	Best	426	538	21282	26130	6528	29368	42214	108263
PPACO	Mean	427	540	21291	26232	6556	29480	42832	108983
	PD_Best(%)	0	0	0	0	0	0	0.44	0.97
	Best	426	538	21282	26328	-	29694	43291	-
EDHACO	Mean	431	545	21355	26873	-	30391	43926	-
(2019)[46]	PD_Best(%)	0	0	0	0.76	-	1.11	3	-
	Best	426	538	21282	-	6570	29533	-	-
PACO-3Opt	Mean	427	539	21327	-	6601	29645	-	-
(2018)[44]	PD_Best(%)	0	0	0	-	0.64	0.56	-	-
	Best	426	538	21282	26130	-	29391	42461	-
PCCACO	Mean	427	539	21383	26241	-	29485	42933	-
(2019)[24]	PD_Best(%)	0	0	0	0	-	0.07	1.03	-
	Best	426	538	21301	-	6538	29468	-	-
PSO-ACO-3Opt	Mean	427	539	21445	-	6564	29646	-	-
(2015)[43]	PD_Best(%)	0	0	0.09	-	0.15	0.34	-	-
	Best	426	538	21283	-	6533	29370	-	-
AS-SA-opt	Mean	-	-	-	-	-	-	-	-
(2018)[38]	PD_Best(%)	0	0	0.005	-	0.08	0.007	-	-
	Best	426	538	21283	-	6531	29372	-	-
HACO	Mean	-	-	-	-	-	-	-	-
(2017)[40]	PD_Best(%)	0	0	0.005	-	0.05	0.01	-	-
	Best	429	-	21753	-	-	-	45280	114158
GA	Mean	432	-	22014	-	-	-	46609	117905
(2015)[15]	PD_Best(%)	0.7	-	2.2	-	-	-	7.7	6.4
	Best	426	540	21282	-	6558	-	-	-
FOA	Mean	427	544	21357	-	6618	-	-	-
(2017)[16]	PD_Best(%)	0	0.3	0	-	0.45	-	-	-
	Best	426	538	21282	26130	6528	29368	-	-
PSO	Mean	426	538	21282	26153	6537	29495	-	-
(2018)[17]	PD_Best(%)	0	0	0	0	0	0	-	-
	Best	429	550	-	26799	-	30161	-	-

Table 16 (continued)

ACO		Eil51	Eil76	KroA100	KroB150	Ch150	KroA200	Lin318	Pr439
ABC	Mean	433	554	-	26988	-	30229	-	-
(2012)[18]	PD_Best(%)	0.15	0.91	-	2.52	-	2.63	-	-

can jump out of the local optimum and get a high-precision solution.

The experimental results show that the solution quality and convergence of the PPACO in this paper are greatly improved for medium scale and large scale TSP instances. The experiment also proves that compared with some single colony ant colony algorithms, multi-colony ant colony algorithms and other intelligent algorithms, the PPACO still has better performance. The future research direction will be a hybrid algorithm, which is generated by the combination of multi-colony ant colony algorithm and game theory, so as to better improve the performance of the algorithm.

Acknowledgments 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.

References

1. Abualigah LM, Khader AT, Hanandeh ES (2018) A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering. Intelligent Decision Technologies 12:3–14
2. Abualigah L, Shehab M, Alshinwan M, Alabool H (2019) Salp swarm algorithm: a comprehensive survey. Neural Comput Applic 32:11195–11215
3. Shehab M, Abualigah L, Al Hamad H, Alabool H, Alshinwan M, Khasawneh AM (2020) Mothflame optimization algorithm: variants and applications, vol 32
4. Shehab M, Alshawabkah H, Abualigah L, Al-Madi N (2020) Enhanced a hybrid moth-flame optimization algorithm using new selection schemes. Eng Comput 36:1–26
5. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671
6. Glover F (1977) Heuristics for integer programming using surrogate constraints. Decis Sci 8(1):156–166
7. Greiner R (1996) PALO: A probabilistic hill-climbing algorithm. Artif Intell 84(1):177–208
8. Whitley D (1994) A genetic algorithm tutorial. Stat Comput 4(2):65–85
9. Schwefel H-P (1977) Evolutionsstrategien für die numerische Optimierung. In: Schwefel H-P (ed) Numerische optimierung von computer-modellen mittels der evolutionsstrategie: mit einer vergleichenden einführung in die hill-climbing- und zufallsstrategie. Basel, Birkhäuser Basel, pp 123–176
10. Xin Y, Yong L, Guangming L (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
11. Dorigo M (1992) Optimization learning and natural algorithms. PhD Thesis, Politecnico di Milano
12. Kennedy J (1944) Eberhart R Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995 1995. pp 1942–1948 vol
13. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
14. Pan W-T (2012) A new Fruit Fly Optimization algorithm: Taking the financial distress model as an example. Knowl-Based Syst 26:69–74
15. Dinh Thanh P, Thi Thanh Binh H, Thu Lam B (2015) New Mechanism of Combination Crossover Operators in Genetic Algorithm for Solving the Traveling Salesman Problem. In: Nguyen V-H, Le A-C, Huynh V-N (eds) Knowledge and Systems Engineering, Cham, 2015. Springer International Publishing, pp 367–379
16. Huang L, Wang G-c, Bai T, Wang Z (2017) An improved fruit fly optimization algorithm for solving traveling salesman problem. Frontiers of Information Technology & Electronic Engineering 18(10):1525–1533
17. Bouzidi M, Riffi ME, Serhir A (2018) Discrete Particle Swarm Optimization for Travelling Salesman Problems: New Combinatorial Operators. In: Abraham A, Haqiq A, Muda AK, Gandhi N (eds) Proceedings of the Ninth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2017), Cham, 2018//. Springer International Publishing, pp 141–150
18. Li L, Cheng Y, Tan L, Niu B (2012) A Discrete Artificial Bee Colony Algorithm for TSP Problem. In: Huang D-S, Gan Y, Premaratne P, Han K (eds) Bio-Inspired Computing and Applications, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg, pp 566–573
19. Dorigo M (1992) Optimization learning and natural algorithms. PhD Thesis, Politecnico di Milano
20. Parvin H, Moradi P, Esmaeili S (2019) TCFACO: Trust-aware Collaborative filtering method based on ant colony optimization. Expert Syst Appl 118:152–168
21. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell 32:112–123
22. Luo YB, Waden YP (2017) The improved ant colony optimization algorithm for MLP considering the advantage from relationship. Math Probl Eng 2017:11
23. Lee J (2017) Heterogeneous-ants-based path planner for global path planning of mobile robot applications. International Journal of Control Automation and Systems 15(4):1754–1769
24. Zhu H, You X, Liu S (2019) Multiple ant colony optimization based on pearson correlation coefficient. IEEE Access 7:61628–61638
25. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics) 26(1):29–41
26. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
27. Stützle T, Hoos HH (2000) MAX–MIN Ant system. Futur Gener Comput Syst 16(8):889–914

28. Wu Y, Ma W, Miao Q, Wang S (2019) Multimodal continuous ant colony optimization for multisensor remote sensing image registration with local search. Swarm and Evolutionary Computation 47:89–95
29. Ratanavilisagul C (2017) Modified Ant Colony Optimization with pheromone mutation for travelling salesman problem. In: 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 27-30 June 2017. pp 411–414
30. Gao W (2016) Premium-penalty ant colony optimization and its application in slope stability analysis. Appl Soft Comput 43:480–488
31. Ye K, Zhang C, Ning J, Liu X (2017) Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems. Inf Sci 406-407:29–41
32. Chen L, Sun H-Y, Wang S (2012) A parallel ant colony algorithm on massively parallel processors and its convergence analysis for the travelling salesman problem. Inf Sci 199:31–42
33. Zhang Q, Zhang C (2018) An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem. Neural Comput & Applic 30(10):3209–3220
34. Shetty A, Shetty A, Puthusseri KS, Shankaramani DR (2018) An Improved Ant Colony optimization Algorithm: Minion Ant(MAnt) and its Application on TSP. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 18-21 Nov. 2018. pp 1219–1225
35. Jun-man K, Yi Z (2012) Application of an improved ant colony optimization on generalized traveling salesman problem. Energy Procedia 17:319–325
36. Xiao J, Li L (2011) A hybrid ant colony optimization for continuous domains. Expert Syst Appl 38(9):11072–11077
37. Dahan F, El Hindi K, Mathkour H, AlSalman H (2019) Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem. Sensors 19(8):1837
38. Yindee O, Arit T (2018) Modified Ant Colony Optimization Algorithm for Multiple-vehicle Traveling Salesman Problems. International Journal of Networked and Distributed Computing 7(1):29–36
39. Kaabachi I, Jriji D, Krichen SA (2017) DSS Based on Hybrid Ant Colony Optimization Algorithm for the TSP. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh LA, Zurada JM (eds) Artificial Intelligence and Soft Computing, Cham, 2017//. Springer International Publishing, pp 645–654
40. Luan J, Yao Z, Zhao F, Song X (2019) A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization. Math Comput Simul 156:294–309
41. Mohsen AM (2016) Annealing ant colony optimization with mutation operator for solving TSP. Computational Intelligence and Neuroscience 2016:13
42. Zhang B, Qi H, Sun SC, Ruan LM, Tan HP (2016) A novel hybrid ant colony optimization and particle swarm optimization algorithm for inverse problems of coupled radiative and conductive heat transfer. Therm Sci 20(2):461–472
43. Mahi M, Baykan Ök, Kodaz H (2015) A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. Applied Soft Computing 30:484–490
44. Gülcü S, Mahi M, Baykan ÖK, Kodaz H (2018) A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. Soft Comput 22(5):1669–1685
45. Wang X, Yang K, Yang L (2018) Application Research of Inner-plant Economical Operation by Multi-colony Ant Optimization. Water Resour Manag 32(13):4275–4295
46. Chen J, You X, Liu S, Li J (2019) Entropy-Based Dynamic heterogeneous ant colony optimization. IEEE Access 7:56317–56328
47. Yang L, Sun X, Zhu A, Chi TH (2017) A multiple ant colony optimization algorithm for indoor room optimal spatial allocation. Isprs International Journal of Geo-Information 6(6):161
48. Tuani AF, Keedwell E, Collett M (2018) H-ACO: A Heterogeneous Ant Colony Optimisation Approach with Application to the Travelling Salesman Problem. In: Lutton E, Legrand P, Parrend P, Monmarche N, Schoenauer M (eds) Artificial Evolution, Cham, 2018. Springer International Publishing, pp 144–161
49. Zhao J, Xue W, Hao C (2018) Heterogeneous Feature Ant Colony Optimization Algorithm Based on Effective Vertexes of Obstacles. In: 2018 Chinese Automation Congress (CAC), 30 Nov.-2 Dec. 2018. pp 30–34
50. Yao Y, Ni Q, Lv Q, Huang K (2015) A novel heterogeneous feature ant colony optimization and its application on robot path planning. In: 2015 IEEE Congress on Evolutionary Computation (CEC), 25-28 May 2015. pp 522–528
51. Xu M, You X, Liu S (2017) A novel heuristic communication heterogeneous dual population ant colony optimization algorithm. IEEE Access 5:18506–18515
52. Wei L, Yuren Z (2010) An Effective Hybrid Ant Colony Algorithm for Solving the Traveling Salesman Problem. In: 2010 International Conference on Intelligent Computation Technology and Automation, 11-12 May 2010. pp 497–500

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Han Pan was born in Anqing, Anhui Province, China in 1994. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include intelligent algorithm, path planning of mobile robot, and embedded system.

Xiaoming You was born in Huaihua, Jiangsu Province, China in 1963. She received the Ph.D. degree in computer science from East China University of Science and Technology in 2007 year. She is a professor and M.S. supervisor at Shanghai University of Engineering Science. Her research interests include swarm intelligent systems, distributed parallel processing, evolutionary computing.

Sheng Liu was born in Daye, Hubei Province, China in 1966. He received the Ph.D. degree in computer science from East China University of Science and Technology in 2008 year. He is a professor and M.S. supervisor at Shanghai University of Engineering Science. His research interests include quantum inspired evolutionary computation, distributed parallel processing, evolutionary computing.

Dehui Zhang was born in Hefei, Anhui Province, China in 1995. She is an M.S. candidate at Shanghai University of Engineering Science. Her research interests include intelligent algorithm, path planning of mobile robot, embedded system.

She is an undergraduate student at suzhou university, suzhou city, anhui province, China. She has won various scholarships and honors. She has published journal articles: Dynamic grouping Ant Colony Algorithm combined with Cat Swarm Optimization, published in Journal of Frontiers of Computer Science and Technology; Multi-Colony Ant Colony Optimization Based on Generalized Jaccard Similarity Recommendation Strategy, published in IEEE Access.