

Received July 21, 2020, accepted August 20, 2020, date of publication August 25, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019445

High-Frequency Path Mining-Based Reward and Punishment Mechanism for Multi-Colony Ant Colony Optimization

HAN PAN¹, XIAOMING YOU¹, AND SHENG LIU²

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

²School of Management, Shanghai University of Engineering Science, Shanghai 201620, China

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

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

ABSTRACT To solve the problem of falling into local optimum and poor convergence speed of traditional ant colony algorithm, this paper proposes a High-frequency path mining-based Reward and Punishment mechanism for multi-colony Ant Colony Optimization (HRPACO). Firstly, the pheromone concentration on the path of effective strong association is rewarded adaptively according to the lift of association rules to accelerate the convergence speed. Secondly, the pheromone concentration on the path of minimum spanning tree is punished adaptively according to the support of association rules to improve the diversity of the colony. The interaction of reward and punishment mechanism can effectively balance the diversity and convergence. Finally, a self-evolutionary mechanism based on Gaussian filter is proposed to adaptively adjust the pheromone concentration by dynamic smoothing of the pheromone matrix, so as to help the colony jump out of the local optimum. The TSP is used to verify the performance of the algorithm. The simulation results show that the proposed algorithm can effectively accelerate the convergence speed and improve the accuracy of solution, especially for large-scale problems. Meanwhile, path planning is used to verify the feasibility of the proposed algorithm. The simulation results show that the algorithm can find an effective and better path even in the environment of complex obstacles.

INDEX TERMS Reward and punishment mechanism, association rules, minimum spanning tree, gaussian filter, path planning.

I. INTRODUCTION

Traveling Salesman Problem is a typical combinatorial optimization problem. It is a concentrated generalization and simplified form of various complex problems in many fields. Besides, the TSP has an important practical and engineering background, and it is widely used in many fields such as transportation, computer network, circuit board design, and logistics distribution. Therefore, any method that simplifies the solution of the problem will be highly valued and concerned. 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, people

The associate editor coordinating the review of this manuscript and approving it for publication was Edith C.-H. Ngai¹.

used the optimal solution algorithm 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 Majorize-Minimization (MM), greedy algorithm, and Minimum Spanning Tree (MST). These approximate algorithms can get a feasible solution that is close to the optimal solution quickly. However, the disadvantage is that the degree of approaching the optimal solution is not satisfactory.

Meta-heuristic algorithms are the improvement of the heuristic algorithm, which is the product of the combination of random algorithms and local search algorithms. They usually do not rely on the specific conditions of some problems, so they can be applied to a broader area. Today, the Meta-heuristic algorithm has been successfully applied in engineering, computer network, biological system modeling,

forecasting, pattern recognition, data clustering, feature selection, and other fields [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], tabu search [6], hill climbing [7], etc. Evolutionary computation and swarm intelligence methods are classifications of population-based methods. Evolutionary computation is a search algorithm based on natural selection and natural genetics, such as Genetic Algorithm (GA) [8], Evolutionary Strategy (ES) [9], Evolutionary Programming (EG) [10]. It has the characteristics of self-organization, self-adaptive, and self-learning. 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 random probability 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]. Swarm intelligence algorithms realize the information exchange and cooperation between individuals and groups. The individual has certain randomness, which keeps the diversity of search directions to a certain extent, and avoids premature convergence and falls into local optimality. The group grasps the direction of optimization on the whole to ensure the convergence of the algorithm. Therefore, in recent years, more and more swarm intelligence algorithms have been used to solve TSP [15]–[20].

The ant colony optimization is a probabilistic algorithm for finding the optimal paths. It was proposed by Dorigo in his doctoral thesis in 1992, 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 [21], Feature selection [22], machine layout problem [23], path planning problem [24], and other fields, and has obtained remarkable results. Ant colony algorithms have become a common method for solving robot path planning problems, and therefore, our research group is studying the use of ant colony algorithms to solve robot path planning problems. The path planning has always been a hot topic and key problem in the field of artificial intelligence research, and it shows great application value. However, TSP also plays an important role as a direction of robot path planning.

Due to the problems of poor convergence and precociousness of traditional ant colony algorithms, many experts and scholars have proposed improved ant colony algorithms. However, the performance of most single colony ant colony algorithms still needs to be improved. The multi-colony ant colony algorithm was found to be superior to the single colony ant colony algorithm in performance. Therefore, the research of the multi-colony algorithm has become an inevitable trend. The detailed development process of the ACO and its main strengths and weaknesses will be discussed in the relevant work.

Inspired by the above analysis, a High-frequency path mining-based Reward and Punishment mechanism for multi-colony Ant Colony Optimization is proposed in this paper. We focus on improving the accuracy of the solution on large-scale problems and balance the diversity and the convergence performance of the algorithm. We selected 13 TSP instances of different scales to verify the performance of HRPACO and compared them with the traditional ant colony algorithm and the improved meta-heuristic algorithm in this field. Meanwhile, four obstacle environments of different complexity were selected to prove the feasibility of HRPACO. Secondly, to illustrate that HRPACO 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 HRPACO is better than several algorithms mentioned in this paper. It can effectively accelerate the convergence speed and obtain more accurate solutions. In addition, HRPACO can find an effective and better solution even in a complex environment. The main contributions of this paper are summarized as follows:

1. The average of similarities between the minimum spanning tree and the optimal paths is used as the evaluation criterion to adaptively adjust the communication frequency of the colony and realize the information sharing among the sub-colonies. This will help to improve the adaptive interaction between colonies.
2. Reward mechanism finds the potential connection between paths according to the lift of association rules, and adaptively rewards the pheromone concentration on the effective strong association path, so as to accelerate the convergence speed. The punishment mechanism evaluates the frequency of path selection according to the support of association rules, and adaptively punishes the pheromone concentration on the path of the minimum spanning tree, so as to improve the diversity. The combination of the two strategies can effectively balance diversity and convergence.
3. A self-evolutionary mechanism based on the Gaussian filter is proposed. When the colony falls into the local optimum, the self-evolutionary mechanism is implemented adaptively according to the self-evolutionary

condition. The mechanism adaptively adjusts the pheromone concentration by dynamic smoothing of the pheromone matrix. It can help the colony to jump out of the local optimum.

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, Minimum spanning tree, Association Rules, Gaussian filter and Grid method. Reward and punishment mechanism based on high-frequency path mining and self-evolutionary mechanism are proposed in Section 4. Section 5 illustrates the experimental results of TSPs, the experimental results of path planning and comparison among different algorithms. Section 6 summarizes our work and describes some of our future directions.

II. THE RELATED WORK

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

Ant Colony Optimization is one of the most effective meta-heuristic algorithms, which simulates the foraging behavior of ants in nature. It has been successfully applied to solve combinatorial optimization problems. In 1992, Dorigo *et al.* proposed the Ant System (AS) [25] inspired by the mechanism of biological evolution. Because 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 continuous expansion of the scale of test cases, the performance has declined seriously. Its main defects are slow convergence and easy to fall into local optimum.

To solve these problems, in 1996, Dorigo proposed the ant colony system (ACS) [26], which is a modified algorithm based on the AS. The new 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, S tutzle and others in the experimental analysis and application research of AS puts forward the Max-Min Ant System (MMAS) [27], MMAS allows only the best Ant in each iteration updating the pheromone trails, and limit the pheromone concentration value of upper and lower. Its purpose is to prevent algorithm premature stop leakage, increase the diversity of the algorithm. However, the traditional ant colony algorithm mentioned above still has some defects such as insufficient convergence, low precision, and easy to fall into local optimum.

A large number of variations of ACO have been presented over the past few years [28]–[37]. 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]. Ye *et al.* proposed to take advantage of search-history information and continually obtains failure experience to guide the ant swarm exploring the unknown space during the

optimization process, 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 to avoid early convergence [32]. To further accelerate the convergence, 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].

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. This algorithm can absorb the advantages of other algorithms so as to obtain better performance [38]–[45]. Xiao *et al.* proposed a hybrid ant colony optimization approach that combines with the continuous population-based incremental learning and the differential evolution for continuous domains. To alleviate the less diversity problem in traditional population-based ant colony algorithms, differential evolution is employed to calculate Gaussian mean values for the next generation [38]. Dahan *et al.* embedded the 3-Opt algorithm into the ant colony algorithm, the number of neighboring nodes that received pheromones varied depending on the quality of the solution compared to obtain high-quality solutions [39]. Yindee *et al.* used four local search strategies in the algorithm: simulated annealing, simulated annealing with similarity measure, 2-opt, and 3-opt, so as to avoid the stagnation of the algorithm [40]. To effectively balance diversity and convergence, Kaabachi *et al.* proposed a new hybrid approach that combines local search with the ant colony optimization algorithm for solving the TSP [41].

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 [46]–[56]. 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. Chen *et al.* used entropy to measure diversity, and the entropy-based allotropic mechanism with three communication strategies can improve the adaptability of the algorithm. Then, the heterogeneous colonies with complementary advantages are proposed to balance the convergence speed and the diversity of the algorithm [48]. Zhu *et al.* proposed a Multiple Ant Colony Optimization based on the Pearson Correlation Coefficient in order to avoid getting trapped in local optimization and enhance the diversity of algorithms [51]. 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 [52]. 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 [53].

The selection of the single colonies that form a multi-colony algorithm is very important. To some extent, using heterogeneous ant optimization algorithms is more possible to avoid premature convergence [50]. We select two classical ant colony algorithms ACS, MMAS 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. The combination of these two single colonies with different characteristics can improve the performance of the algorithm. To achieve a better balance between intensification and diversification, the winner ant is awarded and the loser ant is punished, according to a feedback mechanism called Rule of Winner and Loser [57]. The best solution is rewarded to enhance the guiding effect of the current optimal solution on subsequent iterations. The penalty for the worst solution is used to reduce the misleading effect of the worst path on subsequent iterations so that the algorithm accelerates the convergence speed to the global optimal solution [58]. Therefore, adding the operation of reward and punishment into the algorithm is beneficial to the performance improvement of the algorithm. Besides, association rules reflect the interdependence and relevance between one thing and other things. If we find out the potential correlation, it will not only improve the speed of the solution, but also improve the accuracy of the solution. Shang *et al.* introduce association rules into the ant colony algorithm to solve TSP. The algorithm finds relation above all cities in the proper size of solutions according to association rule [59]. Gao *et al.* tried to combined ACO and the strong association rules to improve the accuracy of the solution [60].

For better performance, the colonies with the above advantages should also be applied. All of the above are what motivate our work.

III. MATERIALS AND METHODS

A. THE PRINCIPLE OF ACS

1) PATH CONSTRUCTION

In ACS algorithm, the ant selects the next node using a state transition rule which is different from AS algorithm. It is a pseudo-random ratio rule controlled by a parameter q_0 . The formula is as follows

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

where q = a random variable, which ranges from 0 to 1. q_0 = an adjustable parameter, which ranges from 0 to 1. i = current node, j = next node, η_{ij} = the reciprocal of the distance between nod i and node, τ_{ij} = the pheromone intensity values between nod i and node, *allowed* = the set of nodes, which is chosen by the ants next. J = a random variable, which is generated from the probability distribution

given in (2):

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

where α determines the importance of the pheromone information; β determines the importance of the heuristic information. The larger their values, the greater their role in the probability of state transfer.

2) PHEROMONE UPDATES

1. Global update rule:

In ACS, global pheromones are updated after all ant cycles have been completed and only the pheromones on the current best path are updated. Using this update method makes the path search more targeted and the ants prefer to search for the optimal path. The update rule is as follow:

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

where $\Delta\tau_{ij} = 1/L_{gb}$, L_{gb} = the current global optimal path length of the algorithm, $\Delta\tau_{ij}$ = increment of pheromone, which is released by the ants on the path from nod i to node j in the current iteration, ρ = the volatility coefficient of the global pheromone. The update of global pheromone is helpful to improve the convergence speed of the algorithm.

2. Local update rule:

After each ant moves from node i to the next node j , the pheromones on the path between the two node are updated according to (4). This pheromone updating method makes ants more likely to choose a different path than the previous one during path construction. The formula of the local update rule is as follows.

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

where ρ = the volatility coefficient of local pheromone, τ_0 = the initial value of the pheromone on each path. The update of local pheromone can prevent algorithm stagnation and increase the diversity of the algorithm.

B. 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:

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

In order 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 $[\tau_{min}, \tau_{max}]$. If the concentration of pheromone on the current edge is higher than τ_{max} , then the concentration of pheromone on the current edge is set to τ_{max} , as shown in the following formula:

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

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

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

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

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

C. DEFINITION AND CONSTRUCTION OF MINIMUM SPANNING TREE

Suppose that the directionless connected band-weight graph $G = \langle V, E, W \rangle$, T is a spanning tree of G . The sum of the rights of the sides of T is called the weight of T , which is denoted as $W(T)$. The spanning tree with the least weight among all spanning trees of G is called the minimum spanning tree of G .

The Prim and Kruskal algorithms are usually used to construct the minimum spanning tree.

The process of the Prim algorithm is described: the vertex is always dominant and the choice of the starting point is arbitrary. The minimum weighted edge is chosen from the starting point to some other point, and then the minimum weighted edge is found at each of the two vertices of this edge. In addition, edges that are already indirectly connected are skipped.

The process of the Kruskal algorithm is described: the edge is always dominant, the minimum weighted edge currently available is always chosen, and each time it is judged whether the two points are already indirectly connected to each other, and if so, this edge is skipped.

D. ASSOCIATION RULES

Association rules reflect the interdependencies and relevance between one thing and other things. It is an important technique in data mining for extracting valuable correlations between data items from large amounts of data.

The process of association rules mining mainly consists of two stages: the first stage must find out all high-frequency itemsets from the data set, that is, the item set whose support is greater than the set minimum threshold is found from the transaction data set. The second stage is to generate association rules from these high-frequency itemsets, that is, to extract the rule whose confidence is higher than the minimum threshold set.

The rules that satisfy the minimum support and minimum confidence are called a ‘‘strong association rule’’. However, in the strong association rule, there are also valid strong association rules and invalid strong association rules. The lift reflects the relevance between the key content of X and Y in association rules, that is, to judge whether it is a valid strong association. If the lift is greater than 1, then the rule ‘‘ $X \rightarrow Y$ ’’ is a valid strong association rule; if the lift is less

than 1, then the rule ‘‘ $X \rightarrow Y$ ’’ is an invalid strong association rule; if the lift is equal to 1, then X and Y are independent of each other and have no relationship.

According to the three metrics of support, confidence and lift, association rules that meet the conditions can be screened out.

- Support: the number of occurrences of several associated data in the dataset as a proportion of the proportion of the total dataset.

$$Support(X, Y) = \frac{P(XY)}{P(All)} \quad (8)$$

- Confidence: the probability that one data appears followed by another, or the conditional probability of the data.

$$Confidence(Y \rightarrow X) = P(X|Y) = P(XY)/P(Y) \quad (9)$$

- Lift: the ratio of the probability of X occurring in the condition containing Y to the probability of X occurring overall.

$$Lift(Y \rightarrow X) = \frac{P(X|Y)}{P(X)} = \frac{Confidence(Y \rightarrow X)}{Support(X)} \quad (10)$$

E. GAUSSIAN FILTER

Under the concept of image processing, Gaussian filter connects image frequency domain processing and time-domain processing, and is used as a low-pass filter to filter out low-frequency energy and to smooth out images.

Gaussian filter is a linear smoothing filter, which is suitable for eliminating Gaussian noise and widely used in the process of image processing noise reduction. Generally speaking, Gaussian filtering is the process of the weighted average of the whole image. The value of each pixel is obtained by the weighted average of itself and other pixel values in the neighborhood. The specific operation of Gaussian filtering is to scan every pixel in the image with a template (or convolution, mask), and use the weighted average gray value of the pixel in the neighborhood determined by the template to replace the value of the central pixel of the template. Gaussian smoothing filter is very effective to suppress the noise which obeys the normal distribution.

F. THE GRID METHOD

The grid method is the representation of maps by coded grids. It marks grids that contain obstacles as obstacle grids, otherwise as free grids, and uses them as the basis for path search.

It is assumed that the workspace of the robot is a finite area on the two-dimensional plane, and the obstacles distributed in the workspace are static, finite, and its location and size are known. The working area is divided into a grid of unit size. If there is no obstacle in the grid, it is called a free grid, which is represented by white and recorded as 0; otherwise, it is called the obstacle grid, which is represented by black and recorded as 1, as shown in Fig. 1 a. When the robot is unobstructed and not at an edge grid, there are eight directions that

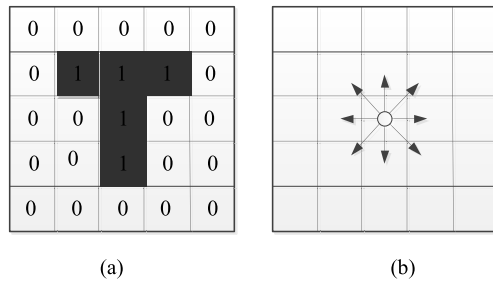


FIGURE 1. Obstacle Representation and Robot movement direction.

can move to the adjacent grid: right, upper right, lower right, left, upper left, lower left, up, down, as shown in Fig. 1 b. A grid coordinate system can be established according to the starting point, target point and obstacles of the robot.

IV. HIGH FREQUENCY PATH MINING-BASED REWARD AND PUNISHMENT MECHANISM FOR MULTI-COLONY ANT COLONY OPTIMIZATION

A. FREQUENCY OF ADAPTIVE COMMUNICATION

For the communication period between colonies, the regular information exchange is carried out every fixed iteration process, which is more direct and easy to achieve. However, frequent communication will disrupt the search progress within a single colony and increase the calculation time. In addition, less communication will make the algorithm close to the single colony algorithm, which does not show the advantages of multi-colony algorithm. Therefore, in this algorithm, the time interval of communication between sub-colonies is not fixed, but changes according to the evolution degree of the colony and the composition of the optimal solution, which is conducive to adaptive adjustment of the frequency of communication between colonies.

There is a close relationship between the minimum spanning tree and the standard optimal solution of TSP. Some studies show that the similarity between the minimum spanning tree and the standard optimal path of TSP instances is as high as 70% ~ 80% [61]. Therefore, when the average value of the similarities between the minimum spanning tree path and the optimal paths of the sub-colonies reaches above a certain threshold, and if the average value remains unchanged for a period of iteration, the sub-colony can be considered to be immersed in local optimum. The current iteration carries out communication between colonies. Thus, the colony can adjust the frequency of communication adaptively and help the algorithm jump out of the local optimum. The iterations selected for communication are as follows:

$$iter_0 = \begin{cases} iter, & \text{if } H_{avg} > h_0 \wedge D(H_{avg}) > w_0 \\ Null, & \text{otherwise} \end{cases} \quad (11)$$

where, $iter_0$ = an iteration that requires communication between colonies, $iter$ = current iteration, H_{avg} = the average similarities between the optimal solutions of the sub-colonies and the path of the minimum spanning tree,

h_0 = the threshold, $D(H_{avg})$ = the number of iterations whose value of H_{avg} remains unchanged and w_0 = the threshold.

If $iter_0$ is not Null in the current iteration, the algorithm falls into the local optimum. At this time, the interaction strategies between colonies need to be implemented in the current iteration. In other words, the frequency of colony communication is determined by the frequency of $iter_0$ production, 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 adjust the communication frequency adaptively to make the algorithm break the stagnation state.

B. REWARD AND PUNISHMENT MECHANISM BASED ON HIGH-FREQUENCY PATH MINING

Because of the positive feedback of the selection strategy of a single colony, the pheromone distribution tends to be consistent with the progress of the algorithm, that is, most pheromones will be distributed in a few paths of the loop, which makes the algorithm inevitably stagnate in the later stage. HRPACO adopts heterogeneous ant colonies with different pheromone updating mechanisms for independent search, with the main body consisting of multiple ACS algorithms and multiple MMAS algorithms. According to the evolutionary degree of the current colony, HRPACO can adaptively select the relevant mechanism to learn, so as to effectively improve the accuracy of the solution. In the early stage of the algorithm, due to the differences of search space and pheromone distribution, the reward and punishment mechanism based on high-frequency path mining is adopted. Firstly, the algorithm punishes the path of the minimum spanning tree according to the support of association rules, so that the pheromones do not gather on these paths excessively. Secondly, according to the lift of association rules, the algorithm strengthens the selection probability of effective strong association path, so as to accelerate the convergence speed. Under the influence of reward and punishment mechanism, it can help the algorithm effectively balance the diversity and convergence. In the later stage of the algorithm, the search regions of the sub-colonies become more and more concentrated due to the convergence of the pheromone distribution regions, so the sub-colonies become more and more similar in search experience and optimal solutions, and interspecies communication does not yield valid experience. Therefore, we adopt a self-evolutionary mechanism. Through the dynamic smoothing of the pheromone matrix, the paths with excessive accumulation of pheromones can adaptively reduce the pheromone concentration, while the pheromone concentration on the nearby path is relatively increased, thus expanding the search space and help the algorithm jump out of the local optimum.

1) REWARD MECHANISM

Association rules reflect the interdependence and relevance between one thing and other things. Association rules are

applied to ant colony algorithm to predict the correlation between paths. There are many similar paths among the optimal solutions of sub-colonies, which means that these frequently occurring paths are likely to be part of the standard optimal solution. More specifically, with the emergence of some frequent paths, another path will follow with a certain probability. This shows that there is a potential correlation between paths. We mine these frequent paths, and then judge the potential relevance between these paths, and finally give rewards, which can not only share the search information among the colonies, but also effectively save the time for the colony to expand the search space.

The frequent itemsets in this paper can be understood as the number of times the path set appears in the optimal solution of the sub-colony greater than the minimum support threshold S_m , while the maximum frequent itemset is defined as the frequent superset that does not contain the current frequent itemsets. We calculate the maximum frequent itemsets G_{max} based on the optimal solutions of the sub-colonies. However, not every optimal solution of sub-colony contains complete set of maximum frequent itemsets. Therefore, the optimal solution of the sub-colony contains part of the set of maximum frequent terms that make up G_a . G_b is the absolute complement of G_a in G_{max} . We look for sets of pathways that are potentially strongly associated with the corresponding G_b based on the G_a of the different sub-colonies. Association rules are the implication of the form $X \rightarrow Y$, which means that Y can be derived from X . So X is G_a , and Y is G_b . We calculate the lift degree according to (5), (6), (7). If the lift is greater than 1, it means that with the emergence of G_a , G_b will also appear with a high probability, so we will reward the path in G_b . However, if the lift degree is less than or equal to 1, then G_a and G_b are invalid strong associations. Next, the lift of subsets of G_b and G_a is calculated until there is a subset that can make the lift greater than 1. The process diagram is as follows:

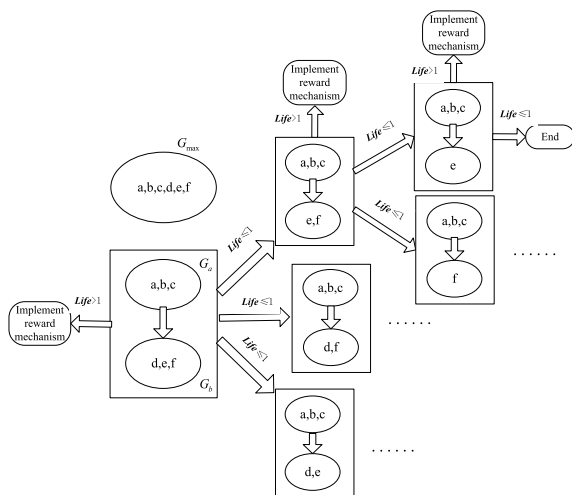


FIGURE 2. Process chart of reward mechanism.

The formula for the award is as follows:

$$G_b^{ij} = G_b^{ij} + L_{ab} \cdot \tau_0 \tag{12}$$

where G_b^{ij} = the path in item set G_b , L_{ab} = the lift between G_a and G_b , and τ_0 = the initial value of pheromone on the path of the city.

The reward operations based on effective strong correlations between paths can effectively reduce the cycle for the colony to explore new paths, allowing the colony to search directionally, thus further accelerating the rate of convergence.

2) PUNISHMENT MECHANISM

The similarity between the minimum spanning tree and the standard optimal solution of TSP instances is 70%~80%, but there are still some paths of the minimum spanning tree that are not part of the standard optimal solution. In the initial iteration of the algorithm, the pheromone concentration between cities is equal, so the pheromone of each path has the same attraction to ants. However, the distance between cities is different, so the probability of ants choosing the next city is dominated by the distance heuristic information between cities, which means that the shorter the distance between cities, the higher the probability of being selected. Unfortunately, the colony has a high probability of choosing the minimum spanning tree path which does not belong to the standard optimal solution, resulting in a pheromone on this path concentrations are high. Thus, when the colony falls into the local optimum, it is likely due to the selection of these minimum spanning tree paths. Therefore, we punish the colony by adaptively reducing the pheromone concentration on the path of the minimum spanning tree, which will reduce the attraction to ants, so as to help the colony have a greater probability to choose other paths and expand the search space. However, the punishing each path of the minimum spanning tree with the same intensity would destroy the previous search experience of the colony, making the probability that the paths of minimum spanning tree that are supposed to form the standard optimal solution are selected excessively low. The significance of support in association rules is to measure the frequency with which an item set appears in the overall set of transactions. When discovering rules, we want to focus on item sets with high frequency, because item sets with low support may appear only by chance, and the itemsets with high support have an expected property. When we apply the support to this paper, the paths with high support are most likely to be a part of the standard optimal solution. Therefore, the punishment formula can be written as follows:

$$\tau_i^{min} = \tau_i^{min} - (1 - S_i^{min}) \cdot \tau_0 \tag{13}$$

where, τ_i^{min} = the pheromone concentration of the minimum spanning tree path i , S_i^{min} = the support of the minimum spanning tree path i , and τ_0 = the initial value of the pheromone on the path.

The value range of S_i^{min} is between 0 and 1, and τ_i^{min} is an increasing function. More specifically, the lower the

support of the minimum spanning tree path, the greater the reduction in its pheromone concentration, thus reducing the probability of selecting an invalid minimum spanning tree path. This also indirectly increases the search probability of other paths and expands the search space, so the diversity of the colony is effectively increased. In addition, in order to prevent too much reduction of pheromone concentration on the path, we introduce the mechanism of the upper and lower limit of the pheromone of MMAS, which allows us to better control the amount of variation in the pheromone concentration.

Reward mechanism provides a convergence function and punishment mechanism provides a diversity function. The mechanisms of rewards and punishments work together and interact with each other to achieve an effective balance between diversity and convergence.

C. SELF-EVOLUTIONARY MECHANISM

Because of the difference of search area and search process among the sub-colonies in the early stage, when the sub-colonies fall into the local optimum, the information exchange is conducive to the mutual learning among the sub-colonies. In this way, the search space can be expanded and the optimization ability can be improved, so that the colony is easier to jump out of the local optimum. However, with the increase of the number of iterations, the search experience between sub-colonies becomes more and more similar, and the paths of the optimal solution are also very similar, which means that communication between subpopulations does not have the desired effect when the population falls into a local optimum again. At this time, another mechanism is adopted to adjust itself according to the information within the colonies. Gaussian filter is used to dynamically smooth the pheromone matrix of sub-colony, and adaptively reduce the high value of the pheromone matrix. Therefore, the colony has a large probability to choose other paths, which jumps out of the local optimum. The condition for selecting the self-evolutionary mechanism is as follows:

$$L_{current}^{opt} = L_{previous}^{opt} \quad (14)$$

where, $L_{current}^{opt}$ = the iteration optimal solution of the colony when $iter_0$ have a new value. $L_{previous}^{opt}$ = the iteration optimal solution of colony for $iter_0$ at the previous old value. $iter_0$ is given by (8). When (11) is satisfied, the current iteration needs to implement a self-evolutionary mechanism.

The two important steps of the Gaussian filter are to find the Gaussian template and then to convolute. The Gaussian kernel is typically an odd-sized Gaussian template. We choose a 3*3 Gaussian template, which is shown as follows:

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

With the Gauss template, we can calculate the dynamically smoothed pheromone matrix. The center point and

Algorithm 1 HRPACO Algorithm for TSP

Input: TSP dataset, Algorithm parameters

Output: Shortest path

1. Initialize the pheromone and parameters
 2. Calculate the distance between cities
 3. While termination condition is not satisfied do
 4. Construct ant solution for ACS with (1), (2)
 5. Construct ant solution for MMAS with (2)
 6. Update pheromone for ACS with (3), (4)
 7. Update pheromone for MMAS with (5), (6), (7)
 8. Calculate the $iter_0$ with (11)
 9. If $iter_0$ is not null
 10. If (14) is not true
 11. Reward the related paths with (8), (9), (10) and (12)
 12. Punish the related paths with (8), (13)
 13. else
 14. Gaussian filter is used to smooth the pheromone matrix
 15. end-if
 16. end-if
 17. NC = NC + 1
 18. END while
 19. Comparing the optimal solutions of ACS and MMAS
 20. Output the minimum solution
-

8 surrounding points multiplied by their own corresponding weight values and summing these values, which is the dynamically smoothed value of the center point. Repeat this process for all points in the pheromone matrix to get the dynamic smoothed pheromone matrix.

Because of the positive feedback mechanism, pheromones tend to focus on the local optimal path, which leads to stagnation. By using Gaussian filter to dynamically smooth the pheromone matrix, the pheromone concentration of the central path is adaptively adjusted according to the pheromone concentrations of the surrounding paths. More specifically, the path with high pheromone concentration adaptively allocates some pheromones to the path with low pheromone concentration around, so that the probability of the colony to choose other paths is greatly increased. In this way, the colony can jump out of the local optimum and find a more accurate solution.

D. ALGORITHM FRAMEWORK

The above is the pseudo-code of the algorithm in this paper. Fig. 3 is the basic framework of the algorithm in this paper. The multi-colony ant colony algorithm proposed in this paper is composed of two types of single colony algorithms: ACS, MMAS. ACS is responsible for accelerating convergence. MMAS is responsible for improving diversity. In the beginning, sub-colonies carry out path optimization and pheromone update operations according to their respective mechanisms. When the search information and the optimal

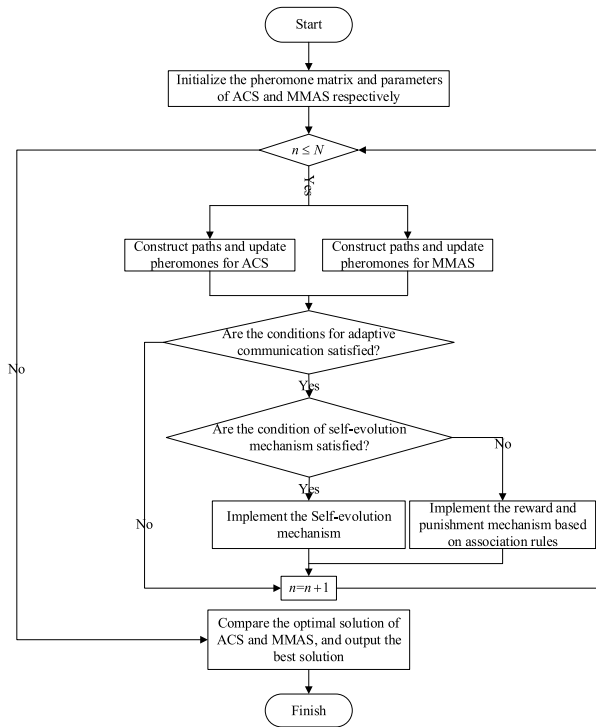


FIGURE 3. Flowchart of the HRPACO.

solution of the sub-colony are quite different, the reward and punishment mechanism based on high-frequency path mining is used to effectively balance the diversity and convergence of the algorithm. When the search direction of the colony tends to be the same, we use the self-evolutionary mechanism to jump out of the local optimum, so as to improve the accuracy of the solution.

E. 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 HRPACO is $n * N * k * m$. Where n = number of sub-colonies, it is a constant, N = maximum number of iterations, k = number of ants per sub-colony, and m = number of cities. So the maximum time complexity of HRPACO 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 HRPACO does not change the maximum time complexity.

V. EXPERIMENTAL SIMULATION AND APPLICATION

The experiment was simulated in MATLAB R2016a on an Intel Core-i5 PC. In order to demonstrate the optimization performance of HRPACO, we selected twelve TSP standard instances from the standard TSPLIB database for systematic analysis. Meanwhile, the classical ACS algorithm and MMAS algorithm are selected to compare the optimization performance with HRPACO. Then, HRPACO is compared with other improved ant colony algorithms and other

intelligent algorithms. Finally, the path planning experiments are carried out on four kinds of simulation maps with different degrees of complexity, and compared with ACS and MMAS. In addition, we also use the map scanned in the real scene to verify the feasibility of the proposed algorithm.

A. PARAMETER SETTING OF THE ALGORITHM

The parameter values of intelligent algorithms are closely related to the actual problem. Researchers usually design experiments to come up with the most suitable set of parameters for their algorithms. In addition, the range of values of the parameters is the same among similar algorithms.

The scientific method of designing experiments should not only reduce the number of experiments as much as possible in the arrangement of experiments, but also make use of the obtained experimental data on the basis of a small number of experiments to analyze the correct conclusions guiding the experiments 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. In order to get a better combination of parameters for HRPACO, $L_{18}(2 \times 3^7)$ orthogonal tables were arranged for experiments. As other orthogonal experiments, the values in this paper are obtained through the preliminary optimization phase. We run 15 times of every program and further calculated the average of TSP eil76.

Based on the experimental results in Tables 1-3, We know that in HRPACO, the best combination of parameters is that $S_m = 0.6, \alpha_A = 1, \beta_A = 4, \rho_A = 0.3, \xi_A = 0.1, \alpha_M = 1, \beta_M = 4, \rho_M = 0.2$.

TABLE 1. Experimental factors and levels of HRPACO.

	S_m	α_A	β_A	ρ_A	ξ_A	α_M	β_M	ρ_M
Level 1	0.4	1	2	0.2	0.1	1	2	0.2
Level 2	0.6	2	3	0.3	0.2	2	3	0.3
Level 3	-	3	4	0.4	0.3	3	4	0.4

S_m = minimum support, ρ_A = the globally updated evaporation factor in the ACS, ξ_A = the locally updated evaporation factor in the ACS, ρ_M = the evaporation factor in the MMAS.

B. STATISTICAL TEST OF THE ALGORITHM

Because ant colony optimization is a probability algorithm, we can only do a limited number of experiments. However, when we analyze the performance difference between algorithms through the experimental results, we cannot judge whether the difference is purely opportunity 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.

TABLE 2. Orthogonal test scheme and test results of HRPACO.

<i>no.</i>	S_m	α_A	β_A	ρ_A	ξ_A	α_M	β_M	ρ_M	<i>results</i>
1	0.4	1	2	0.2	0.1	1	2	0.2	545.5
2	0.4	1	3	0.3	0.2	2	3	0.3	547.4
3	0.4	1	4	0.4	0.3	3	4	0.4	544.2
4	0.4	2	2	0.2	0.2	2	4	0.4	544.6
5	0.4	2	3	0.3	0.3	3	2	0.2	553.8
6	0.4	2	4	0.4	0.1	1	3	0.3	543.1
7	0.4	3	2	0.3	0.1	3	3	0.4	548.8
8	0.4	3	3	0.4	0.2	1	4	0.2	540.8
9	0.4	3	4	0.2	0.3	2	2	0.3	554.9
10	0.6	1	2	0.4	0.3	2	3	0.2	545.1
11	0.6	1	3	0.2	0.1	3	4	0.3	543.7
12	0.6	1	4	0.3	0.2	1	2	0.4	545.8
13	0.6	2	2	0.3	0.3	1	4	0.3	540.4
14	0.6	2	3	0.4	0.1	2	2	0.4	553.6
15	0.6	2	4	0.2	0.2	3	3	0.2	549.9
16	0.6	3	2	0.4	0.2	3	2	0.3	555.1
17	0.6	3	3	0.2	0.3	1	3	0.4	540.5
18	0.6	3	4	0.3	0.1	2	4	0.2	541.3

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

Friedman test only focuses on whether there is a significant difference 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

TABLE 3. Analysis of test results of HRPACO.

<i>K</i>	S_m	α_A	β_A	ρ_A	ξ_A	α_M	β_M	ρ_M
K_1	4923.1	3271.7	3279.6	3279.1	3276.0	3256.1	3308.7	3276.4
K_2	4915.4	3285.4	3279.8	3277.5	3283.6	3286.9	3274.8	3284.6
K_3	-	3281.4	3279.2	3281.9	3278.9	3295.5	3255.0	3277.5
k_1	547.0	545.3	546.6	546.5	546.0	542.7	551.5	546.1
k_2	546.2	547.6	546.6	546.2	547.3	547.8	545.8	547.4
k_3	-	546.9	546.5	547.0	546.5	549.3	542.5	546.3
<i>max</i>	547.0	547.6	546.6	547.0	547.3	549.3	551.5	547.4
<i>min</i>	546.2	545.3	546.5	546.2	546.0	542.7	542.5	546.1
<i>range</i>	0.8	2.3	0.1	0.8	1.3	6.6	9	1.3
<i>scheme</i>	<i>Level 2</i>	<i>Level 1</i>	<i>Level 3</i>	<i>Level 2</i>	<i>Level 1</i>	<i>Level 1</i>	<i>Level 3</i>	<i>Level 1</i>

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

the large-scale TSP instance lin318, the medium-scale TSP instance kroB150, 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 4 is $p = 0 < 0.05$, so the decision is to reject the null hypothesis. In other words, the performance of the four algorithms is significantly different. It can be seen from Fig. 4 that the mean rank of HRPACO is 1.02, the mean rank of ACS is 2.07, the mean rank of EDHACO is 2.95, and the mean rank of MMAS is 3.97. Pairwise comparisons are needed because of the difference in response rates at different frequencies. The results of the pairwise comparison are shown in Table 5. As can be seen from Table 5, the Adj. Sig of HRPACO and ACS is $0.01 < 0.05$. The Adj. Sig of HRPACO and MMAS was $0 < 0.05$. The Adj. Sig of HRPACO and EDHACO was $0 < 0.05$. In conclusion, HRPACO is different from ACS, MMAS, and EDHACO. In other words, the performance comparison between HRPACO and other algorithms has statistical significance in the following experiments.

TABLE 4. Hypothesis test summary.

Null Hypothesis	Test	Sig.	Decision
The distributions of HRPACO, ACS, EDHACO and MMAS 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.

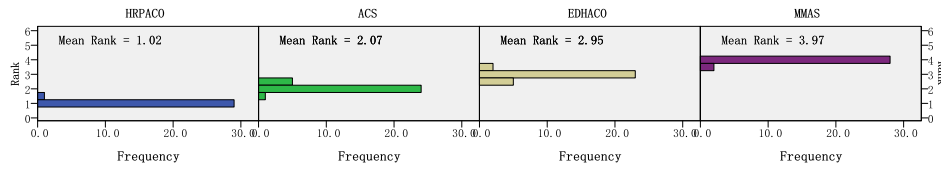


FIGURE 4. Related-Samples Friedman's Two-Way Analysis of Variance by Rank.

TABLE 5. Multiple comparison.

Sample1-Sample	Test Statistic	Std Error	Std. Test Statistic	Sig.	Adj.Sig
HRPACO-ACS	-1.050	0.333	-3.150	0.002	0.010
HRPACO-EDHACO	-1.933	0.333	-5.800	0	0
HRPACO-MMAS	-2.950	0.333	-8.850	0	0
ACS-EDHACO	-0.883	0.333	-2.650	0.08	0.048
ACS-MMAS	-1.900	0.333	-5.700	0	0
EDHACO-MMAS	-1.017	0.333	-3.050	0.002	0.014

C. PERFORMANCE TEST OF HRPACO

In this section, the first part shows the performance comparison between HRPACO and traditional ant colony algorithms in TSP. In the second part, the effectiveness of the mechanism in HRPACO was analyzed. In the third part, HRPACO is compared with the latest improved ant colony algorithm in TSP to demonstrate the performance advantages of HRPACO.

1) COMPARATIVE ANALYSIS OF HRPACO AND TRADITIONAL ANT COLONY ALGORITHMS

To compare the performance of ACS, MMAS, and HRPACO in multiple directions, 12 TSP instances of different scales were selected in this paper. Each TSP instance is executed 15 times. Each experiment was performed by 2,000 iterations. The best solution over all executions (Best), the average solution (Mean), standard deviation (dev), the minimum error rate (Error rate), convergence iteration (Convergence) are applied to evaluate the performance. The experimental data is in Table 6. The standard deviation is calculated as

$$dev = \sqrt{\frac{1}{N} \sum_{i=1}^N (L_i - L_r)^2} \tag{15}$$

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

It can be seen from Table 6 that HRPACO is superior to ACS and MMAS in all aspects of performance. HRPACO is a multi-colony algorithm composed of these two single colonies. To some extent, it also shows that the multi-colony algorithm can make full use of the advantages of a single colony to improve the accuracy of the solution and balance the diversity and convergence. Table 6 can be analyzed in more detail. For small TSP instances with less than 100 cities,

MMAS did not find the standard optimal solutions, but HRPACO and ACS can easily find the standard optimal solutions. In addition, HRPACO can find the standard optimal solutions faster than ACS. This indicates that the reward and punishment mechanism can bring strong convergence ability to HRPACO. In the medium scale and large scale TSP instances with more than 100 cities, ACS and MMAS can hardly find the standard optimal solution, but HRPACO can still find the standard optimal solution, such as kroB150, ch150, kroA200, pr226 and pr264. In pr226 and pr264, the value of HRPACO is larger in the ‘‘convergence’’ column because HRPACO keeps jumping out of the local optimum at a later stage until an optimal solution is found. For lin318, fl417 and pr439, although HRPACO does not find the standard optimal solution, the error rate of the optimal solution obtained by HRPACO is within 1%. These results prove that the self-evolutionary mechanism in the proposed algorithm can effectively improve the accuracy of the solution. On the other hand, mean and dev of HRPACO are smaller than those of ACS and MMAS in TSP instances of different scales, which shows that HRPACO has the ability to obtain stable and high-quality solutions.

The error rate of different scale TSP instances is more visually shown in Fig. 5. We can clearly see that the curve of HRPACO is completely inside the curve of MMAS and ACS, which shows that the error rate of the solution HRPACO is lower than that of ACS and MMAS in different scale TSP instances. It is further proved that HRPACO can improve the accuracy of the solution.

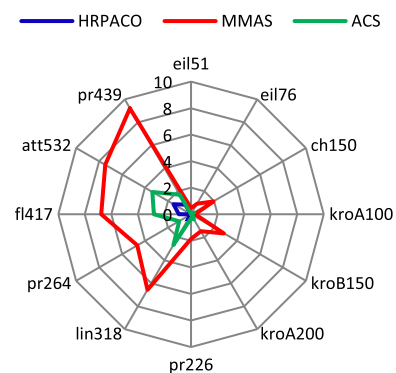


FIGURE 5. Comparison of error rates of ACS, MMAS, and HRPACO.

The convergence curves of different scales of instances are also depicted in Fig. 6, where we take the instance eil51, ch150, pr264 and pr439 as examples to illustrate the

TABLE 6. Performance comparison of HRPACO, ACS, MMAS in different TSP Instances.

TSP 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
		HRPACO	426	427	0.49	0	128
Eil76	538	ACS	538	543	4.58	0	1528
		MMAS	543	550	3.46	0.92	1765
		HRPACO	538	539	0.46	0	260
kroA100	21282	ACS	21282	21316	43.53	0	1538
		MMAS	21349	21568	151.51	0.31	1990
		HRPACO	21282	21296	15.26	0	312
kroB150	26130	ACS	26161	26341	90.22	0.1	923
		MMAS	26878	27197	236.03	2.86	1953
		HRPACO	26130	26218	76.23	0	815
ch150	6528	ACS	6543	6577	21.86	0.22	1125
		MMAS	6655	6744	66.59	1.94	1481
		HRPACO	6528	6540	8.12	0	1075
kroA200	29368	ACS	29495	29626	138.85	0.43	1297
		MMAS	29803	30359	353.64	1.48	1424
		HRPACO	29368	29424	37.69	0	1444
pr226	80369	ACS	80618	80881	337.15	0.3	1332
		MMAS	81852	82531	409.15	1.84	1148
		HRPACO	80381	80502	92.18	0	1801
pr264	49135	ACS	49645	50335	881.84	1.03	1773
		MMAS	51430	52662	621.24	4.67	1741
		HRPACO	49135	49204	71.60	0	1459
lin318	42029	ACS	43139	43650	297.14	2.64	1921
		MMAS	44794	45285	534.43	6.57	1881
		HRPACO	42335	42687	90.68	0.73	1559
fl417	11861	ACS	12193	12330	254.28	2.8	1989
		MMAS	12664	13116	254.267	6.77	1890
		HRPACO	11969	12001	13.66	0.91	353
pr439	107217	ACS	109037	110650	4540.90	1.7	1906
		MMAS	117104	122830	6935.30	9.22	1808
		HRPACO	108132	109089	680.54	0.85	1928
att532	86729	ACS	89652	91044	1487.30	3.37	1986
		MMAS	93211	95398	2861.00	7.47	1878
		HRPACO	88056	90070	442.84	1.53	1722

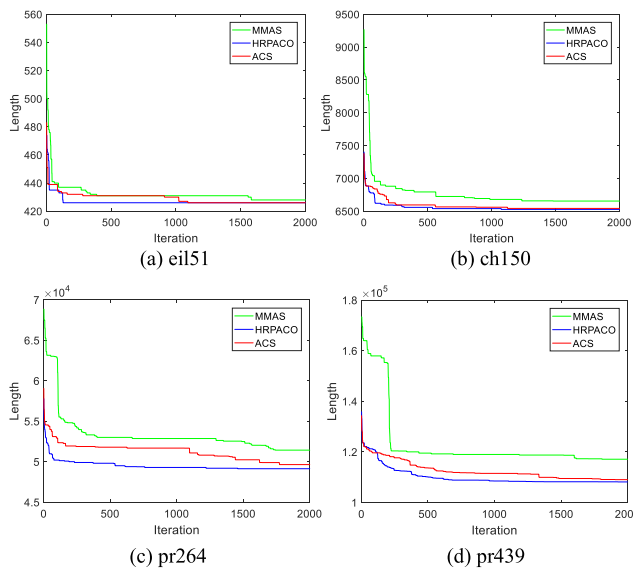


FIGURE 6. Comparison of the convergence of ACS, MMAS, and HRPACO.

convergence ability of our proposed algorithm. As can be seen in Fig. 6, HRPACO shows faster convergence speed than ACS and MMAS in these four TSP instances of different

scales. In addition, Fig. 6 also shows that in ch150, pr264 and pr439, HRPACO can find a better solution than ACS and MMAS. These results show that the algorithm has good global optimization ability and convergence. This also further explains the advantages of multi-colony algorithms and the effectiveness of the interaction mechanism proposed in this paper.

To verify the authenticity of the optimal solution obtained by the algorithm, Fig. 7 illustrates the tours of optimal solutions found by our algorithm in several TSP instances.

2) PERFORMANCE ANALYSIS OF MECHANISMS IN HRPACO

In this subsection, the influence of Reward and punishment mechanism based on high-frequency path minin and the self-evolutionary mechanis is evaluated. The independent experiments have been repeated 15 times in pr264 and lin318, and the performance of mechanism can be discussed depending on the experimental results. The variable controlling approach can be used to validate the performance of each mechanism and their contribution in HRPACO. The experimental analysis is carried out from the following aspects: the optimal solution (Best), the average solution (Mean), the minimum error rate (Error rate), convergence

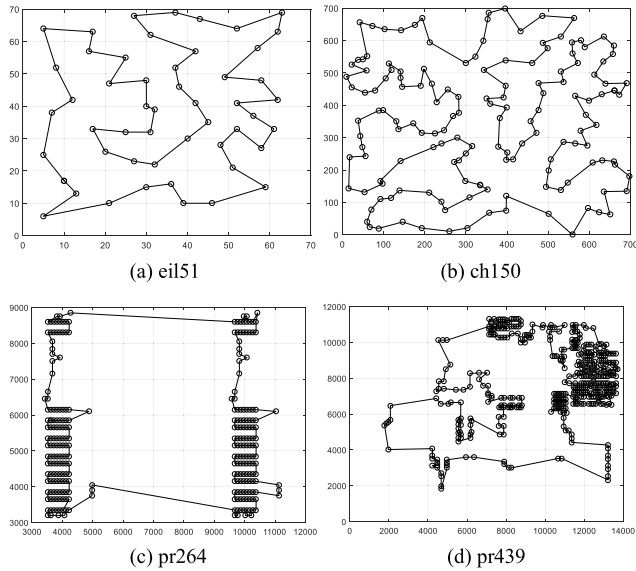


FIGURE 7. Best tours for each TSP instance found by HRPACO.

TABLE 7. Performance analysis of algorithms with different mechanisms.

Instances	ACO	Best	Mean	Error rate(%)	Convergence
Pr264	HRPACO	49135	49204	0	1459
	HRPACO-1	49243	49347	0.22	1726
	HRPACO-2	49368	49497	0.47	1539
	HRPACO-3	49444	49704	0.63	1670
	ACS	49645	50335	1.03	1773
	MMAS	51430	52662	4.67	1741
Lin318	HRPACO	42335	42687	0.73	1559
	HRPACO-1	42589	42817	1.33	1955
	HRPACO-2	42774	42941	1.77	1644
	HRPACO-3	42948	43131	2.19	1612
	ACS	43139	43650	2.64	1921
	MMAS	44794	45285	6.57	1881

iteration (Convergence). The experimental data is in Table 7. HRPACO-1 is an improved multi-colony algorithm with self-evolutionary mechanism but lack of reward and punishment mechanism. HRPACO-2 is an improved multi-colony algorithm with reward and punishment mechanism but lack of self-evolutionary mechanism. HRPACO-3 is an improved multi-colony algorithm without these two communication mechanisms.

First, each mechanism is analyzed to prove the promotion of the algorithm. It can be seen from Fig. 8 that the convergence curves of the four improved algorithms are faster than those of ACS and MMAS in pr264 and lin318. In addition, it can be seen from Table 7 that the optimal and average solutions of the four improved algorithms are still better than those of ACS and MMAS in pr264 and lin318. These results show that the reward and punishment mechanism and

self-evolutionary mechanism are effective in optimizing the algorithm. The performance of multi-colony algorithm without communication mechanism is also improved compared with single colony.

Next, the main role of each mechanism is analyzed in detail. As can be seen in Table 7, the optimal and average solutions of HRPACO-3 are the worst of the four improved algorithms, because this is only the superposition of simple single colony. In addition, the optimal and average solutions of HRPACO-1 are better than those of HRPACO-2 and HRPACO-3 in pr264 and lin318. This shows that the self-evolutionary mechanism can help the algorithm to jump out and get a more accurate solution. As can be seen in Fig. 8, the convergence speed of HRPACO-2 is faster than that of HRPACO-1 and HRPACO-3 in pr264 and lin318. At the same time, the convergence accuracy of HRPACO-2 is better than that of HRPACO-3. These results show that the reward and punishment mechanism can effectively balance diversity and convergence. Finally, Fig. 8 and Table 7 show that HRPACO is better than HRPACO-1, HRPACO-2 and HRPACO-3 in all metrics. Therefore, the reward and punishment mechanism and the self-evolutionary mechanism improve the performance of the algorithm together, and make HRPACO have the high precision solution and strong convergence speed.

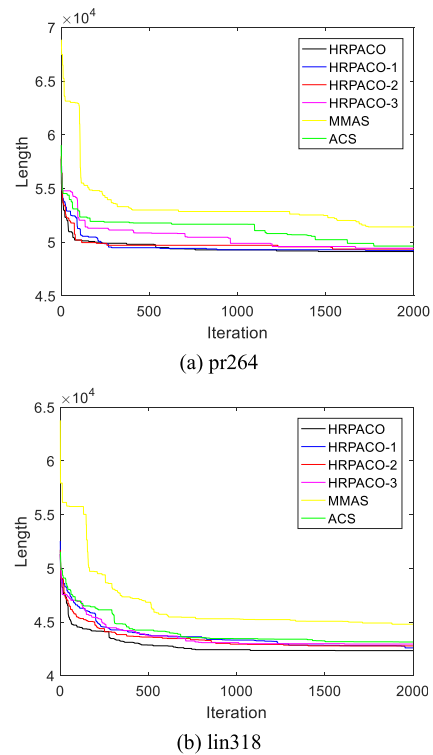


FIGURE 8. Comparison of convergence of algorithms with different mechanisms.

3) COMPARISON WITH THE LATEST IMPROVED ALGORITHMS IN TSP

To further illustrating the performance of the proposed algorithm, HRPACO is also compared with other optimization

TABLE 8. Comparison with other optimization algorithms in TSP.

ACO		Eil51	Eil76	KroA100	KroB150	Ch150	KroA200	Lin318	Pr439
	Opt	426	538	21282	26130	6528	29368	42029	107217
HRPACO	Best	426	538	21282	26130	6528	29368	42335	108132
	Mean	427	539	21296	26218	6540	29424	42687	109089
	PD_Best(%)	0	0	0	0	0	0	0.73	0.85
EDHACO (2019)[48]	Best	426	538	21282	26328	-	29694	43291	-
	Mean	431	545	21355	26873	-	30391	43926.3	-
	PD_Best(%)	0	0	0	0.76	-	1.11	3.00	-
PACO-3Opt (2018)[46]	Best	426	538	21282	-	6570	29533	-	-
	Mean	427	539	21327	-	6601	29645	-	-
	PD_Best(%)	0	0	0	-	0.64	0.56	-	-
PCCACO (2019)[51]	Best	426	538	21282	26130	-	29391	42461	-
	Mean	427	539	21383	26241	-	29485	42933	-
	PD_Best(%)	0	0	0	0	-	0.07	1.03	-
HMMA (2015)[20]	Best	438	549	21371	26585	6654	29999	45349	114094
	Mean	450	558	21425	26971	6749	31405	46548	116885
	PD_Best(%)	2.81	2.04	0.41	1.74	1.93	2.14	7.89	6.41
AS-SA-opt (2018)[40]	Best	426	538	21283	-	6533	29370	-	-
	Mean	-	-	-	-	-	-	-	-
	PD_Best(%)	0	0	0.005	-	0.08	0.007	-	-
HACO (2017)[41]	Best	426	538	21283	-	6531	29372	-	-
	Mean	-	-	-	-	-	-	-	-
	PD_Best(%)	0	0	0.005	-	0.05	0.01	-	-
DSMO (2020)[19]	Best	428	558	21298	26601	-	30481	44118	112105
	Mean	436	572	22024	27575	-	31828	45460	116379
	PD_Best(%)	0.47	3.71	0.07	1.80	-	3.70	4.97	4.55
FOA (2017)[16]	Best	426	540	21282	-	6558	-	-	-
	Mean	427	544	21357	-	6618	-	-	-
	PD_Best(%)	0	0.37	0	-	0.45	-	-	-
ABC (2012)[18]	Best	429	550	-	26799	-	30161	-	-
	Mean	433	554	-	26988	-	30229	-	-
	PD_Best(%)	0.15	0.91	-	2.52	-	2.63	-	-

algorithms. The selected algorithms have been released in the last few years and have various levels of performance improvements. Besides, the criteria for comparing algorithms selected in this paper can be illustrated by the determination of parameters. The parameter values of the smart algorithm are closely related to the actual problem and the size of the problem. The author usually uses experimental tests to find the best set of parameters to achieve the best results. Moreover, the range of parameters options is the same between similar algorithms. The parameter values of the selected similar algorithms in this paper are within the same range. The selection of TSPs is based on the same TSP instances between HRPACO and the comparison algorithms. Therefore, the comparison among the optimization algorithms in this paper is fair. Table 8 shows the data of HRPACO and other improved algorithms. EDHACO, PACO-3Opt, and PCCACO are improved multi-colony ant colony algorithms. HMMA, AS-SA-Opt, and HACO are hybrid algorithms based on ant colony algorithm. DSMO, FOA, 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.

It can be seen from Table 8 that HRPACO can obtain higher precision solution than other optimization algorithms. In small scale and medium scale TSP instances with less than 300 cities, HRPACO can find the standard optimal solution. For example, in ch150 and kroA200, only HRPACO can find the standard optimal solution. In addition, for large scale TSP instances with more than 300 cities, the solution obtained by HRPACO is closer to the standard optimal solution, and the error rate is basically kept within 1%.

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

D. APPLICATION RESEARCH OF THE ALGORITHM

1) ANALYSIS OF SIMULATION RESULTS OF PATH PLANNING
In this section, the algorithm proposed in this paper will carry out simulation experiments in different complex obstacle environments (20×20 , 40×40 , 70×70), and also compare with ACS.

The reasonable environment representation and appropriate search algorithm can plan a more satisfactory path with less time cost. The methods of environment modeling usually include visibility graph, free space method, Maklink diagram, grid method and Voronoi diagram. Because of the advantages of the grid method, such as high precision and easy to implement, this paper will use the most classical grid method.

The results of Fig. 9 show that the convergence speed of HRPACO is obviously faster than that of ACS in a simple obstacle environment of 20×20 scale (starting point $s = 1$, ending point $G = 400$), although two algorithms can reach the ending point in a shorter path. The results of Fig. 10 and Fig. 11 show that the path chosen by HRPACO is shorter than that of ACS in the complex obstacle environment of 40×40 (starting point $s = 1$, ending point $G = 1600$), 70×70 (starting point $s = 1$, ending point $G = 2500$), which shows that the optimization ability of HRPACO is better than that of ACS. In addition, Fig. 12 shows the comparison results of simulation experiments under the 70×70 complex obstacle environment when the starting point and the ending point are special locations (the starting point $S = 4003$ and the ending point $G = 1463$). The results show that the optimization ability and convergence speed of HRPACO are still better than ACS.

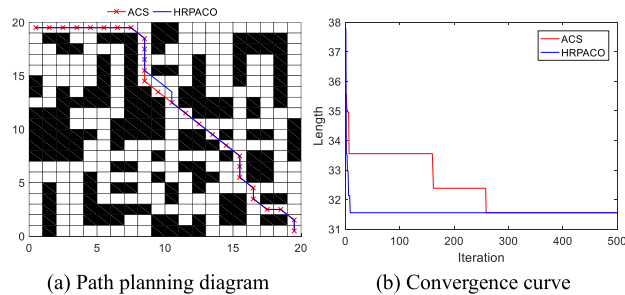


FIGURE 9. Obstacle environment (20×20).

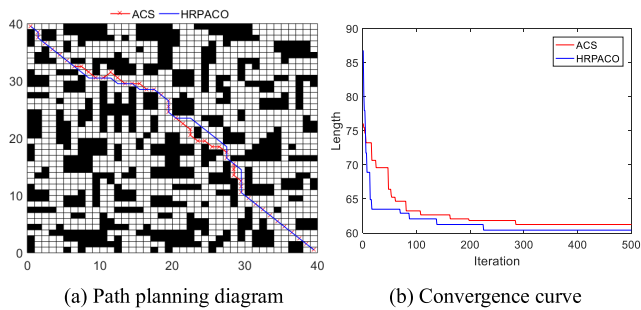


FIGURE 10. Obstacle environment (40×40).

Table 9 shows the experimental data obtained from 20 runs of HRPACO and ACS in each obstacle environment. The experimental analysis is carried out from the following aspects: the optimal solution (Best), the average solution (Mean), standard deviation (dev), the convergence iteration (Convergence). The numbers in parentheses after the scale of the obstacle are the start and end points. It can be seen from

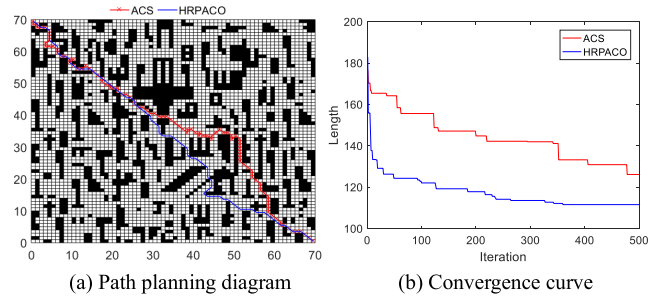


FIGURE 11. Obstacle environment (70×70).

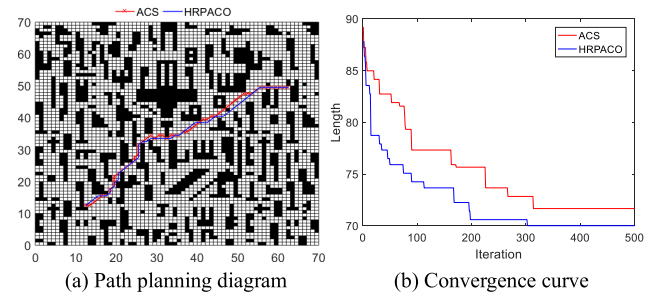


FIGURE 12. Obstacle environment for special starting and ending point (70×70).

TABLE 9. Performance comparison of two algorithms in different environments.

Obstacle environment	Algorithm	Best	Mean	Dev	Convergence
20×20 (1,400)	HRPACO	31.55	32.42	0.91	9
	ACS	31.55	32.78	0.92	259
40×40 (1,1600)	HRPACO	60.42	61.80	1.13	225
	ACS	61.25	63.27	1.23	285
70×70 (1,4900)	HRPACO	111.53	115.89	3.33	360
	ACS	126.12	134.52	5.30	478
70×70 (4003,1463)	HRPACO	70.01	70.97	0.62	303
	ACS	71.66	74.19	1.23	314

Table 9 that with the improvement of the complexity of the obstacle environment, the performance of HRPACO is still better than that of ACS in all metrics. The comprehensive experimental results show that the path planning ability of HRPACO is stronger and it can reach the destination more quickly and accurately in the map of different scales and different complex obstacles. In addition, its performance will not be affected by the start and end points of different positions, and its stability is high.

2) RESEARCH ON THE PRACTICAL APPLICATION OF THE ALGORITHM

In order to reflect the effect of the algorithm proposed in autonomous path planning on static map, the actual topographic map of the experimental environment in this paper is established, as shown in Fig. 13 a. The PGM version of the

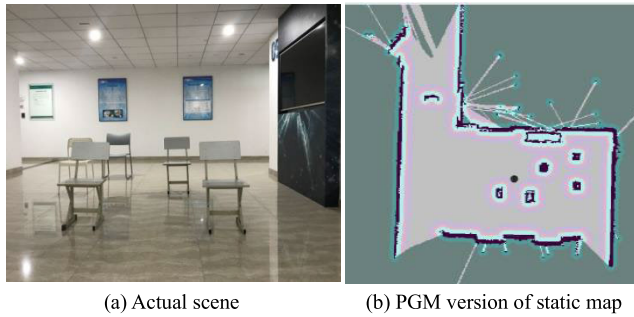


FIGURE 13. Real maps of path planning.

environment map can be obtained by using the map building algorithm in ROS for actual mapping. The black border of the outer part of the map is a wall or a door, and the black square in the middle is the set obstacle, as shown in Fig. 13 b. Then the map is transformed into grid map as shown in Fig. 14, which can be better simulated in MATLAB, so as to compare the algorithm proposed in this paper with ACS.

The starting point coordinate of the path planning is (0, 20), and the target point coordinate is (60, 83). ACS and HRPACO are used for path planning. Fig. 14 shows the simulation results of HRPACO and ACS. ACS can smoothly bypass obstacles in path planning, but the planned path has many setbacks, resulting in path redundancy. However, the path chosen by HRPACO is better than ACS.

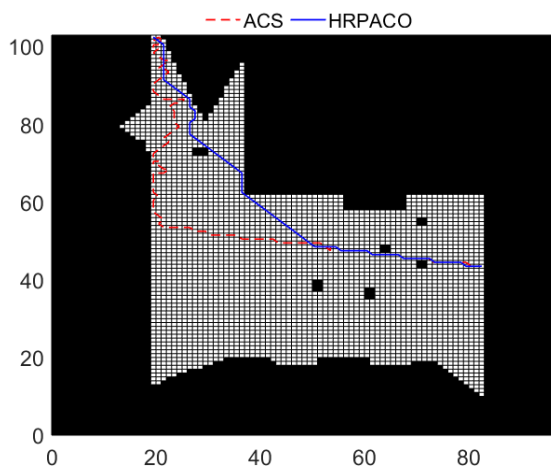


FIGURE 14. Paths of HRPACO and ACS in the grid map.

To sum up, even in the complex real environment, when using HRPACO for path planning, the robot can effectively avoid obstacles and accurately reach the designated destination. Therefore, HRPACO is feasible and effective in robot path planning.

VI. CONCLUSION

The traditional ant colony algorithm is easy to fall into the problem of local optimization, lack of guidance and poor self-adaptability. Therefore, this paper proposes a multi-colony

algorithm based on reward and punishment mechanism of frequent path mining, which is composed of ACS and MMAS. Firstly, the evolutionary degree of the colony is judged according to the similarity between the minimum spanning tree and the optimal solution, so as to adjust the frequency of colony communication adaptively. Then, in the early stage of the algorithm, due to the difference of search space and search experience between colonies, choosing the reward and punishment mechanism based on frequent path mining can effectively improve the performance of the algorithm. More specifically, the path on the minimum spanning tree is punished according to the support in the association rules, so as to reduce the pheromone accumulated on the corresponding path and promote the colony to expand the search space. The association rules expose the potential connections between the paths. According to the lift of association rules, reward the path of effective strong association, so that the colony can save search time and accelerate convergence speed. The reward and punishment mechanisms cooperate with each other, effectively balancing the diversity and convergence of the algorithm.

In the later stage of the algorithm, because the optimal solutions of sub-colonies are more and more similar, the information exchange between colonies cannot achieve the expected effect. Therefore, we adopt the self-evolutionary mechanism, and the Gaussian filter is used to dynamically smooth the pheromone matrix of the sub-colony, so that the colony can jump out of the local optimum quickly and get a more accurate solution.

The experimental results of TSP show that the convergence speed and precision of HRPACO are better than those of traditional ant colony algorithms, other improved ant colony algorithm and other optimization algorithms, especially for large-scale problems. The results of path planning show that the algorithm proposed can be used not only in the general obstacle environment, but also in special obstacle environment. Due to the long computation time in solving complex optimization problems, the HRPACO algorithm needs to be further investigated to reduce the computation time. In order to make the algorithm better applied in the actual scene, the next step is to discuss the multi-objective path planning problem.

REFERENCES

- [1] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering," *Intell. Decis. Technol.*, vol. 12, no. 1, pp. 3–14, 2018.
- [2] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, Nov. 2019.
- [3] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, and A. M. Khasawneh, "Moth-flame optimization algorithm: Variants and applications," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 9859–9884, Oct. 2019.
- [4] M. Shehab, H. Alshawabkha, L. Abualigah, and A. M. Nagham, "Enhanced a hybrid moth-flame optimization algorithm using new selection schemes," *Eng. Comput.*, vol. 36, pp. 1–26, Feb. 2020.

- [5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, p. 671, 1983.
- [6] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decis. Sci.*, vol. 8, no. 1, pp. 156–166, Jan. 1977.
- [7] R. Greiner, "PALO: A probabilistic hill-climbing algorithm," *Artif. Intell.*, vol. 84, nos. 1–2, pp. 177–208, Jul. 1996.
- [8] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.
- [9] H.-P. Schwefel, "Evolutionsstrategien für die numerische optimierung," in *Numerische Optimierung von Computer-Modellen Mittels der Evolutionsstrategie: Mit Einer Vergleichenden Einführung in Die Hill-Climbing- und Zufallsstrategie*, H.-P. Schwefel, Ed. Basel, Switzerland: Birkhäuser Basel, 1977, pp. 123–176.
- [10] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
- [11] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Milan, Italy, 1992.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.
- [13] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.
- [14] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012.
- [15] P. D. Thanh, H. T. T. Binh, and B. T. Lam, "New mechanism of combination crossover operators in genetic algorithm for solving the traveling salesman problem," in *Knowledge and Systems Engineering*. Cham, Switzerland: Springer, 2015, pp. 367–379.
- [16] L. Huang, G.-C. Wang, T. Bai, and Z. Wang, "An improved fruit fly optimization algorithm for solving traveling salesman problem," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 10, pp. 1525–1533, Oct. 2017.
- [17] M. Bouzidi, M. E. Riffi, and A. Serhir, "Discrete particle swarm optimization for travelling salesman problems: New combinatorial operators," in *Proc. 9th Int. Conf. Soft Comput. Pattern Recognit. (SoCPar)*. Cham, Switzerland: Springer, 2018, pp. 141–150.
- [18] L. Li, Y. Cheng, L. Tan, and B. Niu, "A discrete artificial bee colony algorithm for TSP problem," in *Bio-Inspired Computing and Applications*. Berlin, Germany: Springer, 2012, pp. 566–573.
- [19] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete spider monkey optimization for travelling salesman problem," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105887.
- [20] W. Yong, "Hybrid max-min ant system with four vertices and three lines inequality for traveling salesman problem," *Soft Comput.*, vol. 19, no. 3, pp. 585–596, Mar. 2015.
- [21] H. Parvin, P. Moradi, and S. Esmaili, "TCFACO: Trust-aware collaborative filtering method based on ant colony optimization," *Expert Syst. Appl.*, vol. 118, pp. 152–168, Mar. 2019.
- [22] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 112–123, Jun. 2014.
- [23] Y. Luo and Y. P. Waden, "The improved ant colony optimization algorithm for MLP considering the advantage from relationship," *Math. Problems Eng.*, vol. 2017, Jan. 2017, Art. no. 3920327.
- [24] J. Lee, "Heterogeneous-ants-based path planner for global path planning of mobile robot applications," *Int. J. Control, Autom. Syst.*, vol. 15, no. 4, pp. 1754–1769, Aug. 2017.
- [25] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
- [26] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
- [27] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.
- [28] Y. Wu, W. Ma, Q. Miao, and S. Wang, "Multimodal continuous ant colony optimization for multisensor remote sensing image registration with local search," *Swarm Evol. Comput.*, vol. 47, pp. 89–95, Jun. 2019.
- [29] C. Ratanavilisagul, "Modified ant colony optimization with pheromone mutation for travelling salesman problem," in *Proc. 14th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Jun. 2017, pp. 411–414.
- [30] W. Gao, "Premium-penalty ant colony optimization and its application in slope stability analysis," *Appl. Soft Comput.*, vol. 43, pp. 480–488, Jun. 2016.
- [31] K. Ye, C. Zhang, J. Ning, and X. Liu, "Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems," *Inf. Sci.*, vols. 406–407, pp. 29–41, Sep. 2017.
- [32] L. Chen, H.-Y. Sun, and S. Wang, "A parallel ant colony algorithm on massively parallel processors and its convergence analysis for the travelling salesman problem," *Inf. Sci.*, vol. 199, pp. 31–42, Sep. 2012.
- [33] Q. Zhang and C. Zhang, "An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem," *Neural Comput. Appl.*, vol. 30, no. 10, pp. 3209–3220, Nov. 2018.
- [34] A. Shetty, A. Shetty, K. S. Puthusseri, and R. Shankaramani, "An improved ant colony optimization algorithm: Minion Ant(MAnt) and its application on TSP," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1219–1225.
- [35] K. Jun-Man and Z. Yi, "Application of an improved ant colony optimization on generalized traveling salesman problem," *Energy Procedia*, vol. 17, pp. 319–325, 2012.
- [36] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with Levy flight," *IEEE Access*, vol. 8, pp. 67205–67213, 2020.
- [37] J. Li, Y. Xia, B. Li, and Z. G. Zeng, "A pseudo-dynamic search ant colony optimization algorithm with improved negative feedback mechanism," *Cognit. Syst. Res.*, vol. 62, pp. 1–9, Aug. 2020.
- [38] J. Xiao and L. Li, "A hybrid ant colony optimization for continuous domains," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11072–11077, Sep. 2011.
- [39] F. Dahan, K. El Hindi, H. Mathkour, and H. AISalman, "Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem," *Sensors*, vol. 19, no. 8, p. 1837, Apr. 2019.
- [40] O. Yindee and T. Arit, "Modified ant colony optimization algorithm for multiple-vehicle traveling salesman problems," *Int. J. Networked Distrib. Comput.*, vol. 7, no. 1, pp. 29–36, Dec. 2018.
- [41] I. Kaabachi, D. Jrijji, and S. Krichen, "A DSS based on hybrid ant colony optimization algorithm for the TSP," in *Artificial Intelligence and Soft Computing*. Cham, Switzerland: Springer, 2017, pp. 645–654.
- [42] J. Luan, Z. Yao, F. Zhao, and X. Song, "A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization," *Math. Comput. Simul.*, vol. 156, pp. 294–309, Feb. 2019.
- [43] A. M. Mohsen, "Annealing ant colony optimization with mutation operator for solving TSP," *Comput. Intell. Neurosci.*, vol. 2016, Oct. 2016, Art. no. 8932896.
- [44] B. Zhang, H. Qi, S.-C. Sun, L.-M. Ruan, and H.-P. Tan, "A novel hybrid ant colony optimization and particle swarm optimization algorithm for inverse problems of coupled radiative and conductive heat transfer," *Thermal Sci.*, vol. 20, no. 2, pp. 461–472, 2016.
- [45] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-Opt algorithms for traveling salesman problem," *Appl. Soft Comput.*, vol. 30, pp. 484–490, May 2015.
- [46] Ş. Gülcü, M. Mahi, Ö. K. Baykan, and H. Kodaz, "A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem," *Soft Comput.*, vol. 22, no. 5, pp. 1669–1685, Mar. 2018.
- [47] X. Wang, K. Yang, and L. Yang, "Application research of inner-plant economical operation by multi-colony ant optimization," *Water Resour. Manage.*, vol. 32, no. 13, pp. 4275–4295, Oct. 2018.
- [48] J. Chen, X.-M. You, S. Liu, and J. Li, "Entropy-based dynamic heterogeneous ant colony optimization," *IEEE Access*, vol. 7, pp. 56317–56328, 2019.
- [49] L. Yang, X. Sun, A. Zhu, and T. Chi, "A multiple ant colony optimization algorithm for indoor room optimal spatial allocation," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 6, p. 161, Jun. 2017.
- [50] M. Xu, X. You, and S. Liu, "A novel heuristic communication heterogeneous dual population ant colony optimization algorithm," *IEEE Access*, vol. 5, pp. 18506–18515, 2017.
- [51] H. Zhu, X. You, and S. Liu, "Multiple ant colony optimization based on Pearson correlation coefficient," *IEEE Access*, vol. 7, pp. 61628–61638, 2019.
- [52] A. F. Tuani, E. Keedwell, and M. Collett, "H-ACO: A heterogeneous ant colony optimization approach with application to the travelling salesman problem," in *Artificial Evolution*. Cham, Switzerland: Springer, 2018, pp. 144–161.

- [53] J. Zhao, W. Xue, and C. Hao, "Heterogeneous feature ant colony optimization algorithm based on effective vertexes of obstacles," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 30–34.
- [54] Y. Yao, Q. Ni, Q. Lv, and K. Huang, "A novel heterogeneous feature ant colony optimization and its application on robot path planning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 522–528.
- [55] L. Meng, X. You, S. Liu, and S. Li, "Multi-colony ant algorithm using both generative adversarial nets and adaptive stagnation avoidance strategy," *IEEE Access*, vol. 8, pp. 53250–53260, 2020.
- [56] K. Yang, X. M. You, S. Liu, and H. Pan, "A novel ant colony optimization based on game for traveling salesman problem," *Appl. Intell.*, pp. 1–14, 2020, doi: 10.1007/s10489-020-01799-w.
- [57] Y. Yan, H.-S. Sohn, and G. Reyes, "A modified ant system to achieve better balance between intensification and diversification for the traveling salesman problem," *Appl. Soft Comput.*, vol. 60, pp. 256–267, Nov. 2017.
- [58] Q. Luo, H. Wang, Y. Zheng, and J. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1555–1566, Mar. 2020.
- [59] G. Shang, Z. Lei, Z. Fengting, and Z. Chunxian, "Solving traveling salesman problem by ant colony optimization algorithm with association rule," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, vol. 3, 2007, pp. 693–698.
- [60] Y. Gao, J.-Q. Hu, and X.-L. Tang, "The application of hybrid ant colony algorithm in association rule mining," in *Proc. 9th Int. Symp. Comput. Intell. Design (ISCID)*, vol. 2, Dec. 2016, pp. 329–333.
- [61] L. Wei and Z. Yuren, "An effective hybrid ant colony algorithm for solving the traveling salesman problem," in *Proc. Int. Conf. Intell. Comput. Technol. Autom.*, vol. 1, May 2010, pp. 497–500.



XIAOMING YOU was born in Huaihua, Jiangsu, China, in 1963. She received the Ph.D. degree in computer science from the East China University of Science and Technology, in 2007. She is currently a Professor and a M.S. degree Supervisor with the Shanghai University of Engineering Science. Her research interests include swarm intelligent systems, distributed parallel processing, and evolutionary computing.



HAN PAN was born in Anqing, Anhui, 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 systems.



SHENG LIU was born in Daye, Hubei, China, in 1966. He received the Ph.D. degree in computer science from the East China University of Science and Technology, in 2008. He is currently a Professor and a M.S. degree Supervisor with the Shanghai University of Engineering Science. His research interests include quantum inspired evolutionary computation, distributed parallel processing, and evolutionary computing.

...